VIRTUAL CELL IN MOBILE COMPUTER COMMUNICATIONS

By

KYUNGSHIK LIM

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
UNIVERSITY OF FLORIDA

1994

ACKNOWLEDGEMENTS

First of all, I would like to thank my advisor, Dr. Yann-Hang Lee, for his thorough guidance and warm encouragement. Without his endless ideas and keen insight into the problems, I would never have accomplished this work. His patience and open-mindedness have also made my academic life very happy and enjoyable.

I would also like to thank Dr. Randy Chow, who introduced me to the SMDS project that partially motivated this work. His lectures on computer networks and operating systems gave me a strong background for this work.

I would like to thank Dr. Richard E. Newman-Wolfe for his helpful discussions and comments on my presentations in the SMDS project and my dissertation.

I would also like to thank my other supervisory committee members, Dr. Panos Livadas and Dr. Scott L. Miller, for their support and advice.

Special thanks go to my twin boys, Jisok Lim and Hyosok Lim, for growing up in good health and character, and my lovely wife, Hyunhee Lim, for her devotional love to my family. I dedicate this work to them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

ALS	Address resolution protocol/Location Server
AMT	Address Mapping Table
ARP	Address Resolution Protocol
ATM	Asynchronous Transfer Mode
BMAC	Base station network Medium Access Control
BS	Base Station
CPS	Cellular Packet Switch
EMT	External Membership Table
GMT	Global Membership Table
IMT	Internal Membership Table
IP	Internet Protocol
IPIP	Internet Protocol within Internet Protocol
ISDN	Integrated Services Digital Networks
LAN	Local Area Network
LLC	Logical Link Control
L3_PDU	Level 3 Protocol Data Unit
MAC	Medium Access Control
MAN	Metropolitan Area Network
MH	Mobile Host
MICP	Mobile Internetworking Control Protocol
MSG	Mobile Support Gateway
MSS	Mobile Support Station
PID	Protocol Identification

PN	Physical Network
RMAC	Radio network Medium Access Control
SMDS	Switched Multimegabit Data Service
SNAP	Sub-Network Access Protocol
SNI	Subscriber Network Interface
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VCARP	Virtual Cell Address Resolution Protocol
VCI	Virtual Circuit Identifier
VCP	Virtual Cell Protocol
VIP	Virtual Internet Protocol
VN	Virtual Network

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

VIRTUAL CELL IN MOBILE COMPUTER COMMUNICATIONS

By

Kyungshik Lim

December 1994

Chairman: Dr. Yann-Hang Lee
Major Department: Computer and Information Sciences

In this research, we design and develop a *virtual cell* approach for the transmission of IP datagrams in mobile computer communications. A virtual cell consists of a group of physical cells whose base stations are implemented by remote bridges and interconnected via high-speed datagram packet-switched networks. Host mobility is supported at the data link layer using the distributed hierarchical location information of mobile hosts. It eliminates the necessity of IP-level mobile host protocols that may interfere with the conventional IP protocol in a practical sense and achieves a logically flexible coverage area according to mobility and communication patterns among physical cells.

Given mobility and communication patterns among physical cells, the problem of deploying virtual cells is transformed to the optimization problem of finding a cover of disjoint clusters of physical cells. The objective is to minimize the total communication cost for the entire system where intercluster communication is more expensive than intracluster communication. Our problem differs from general graph partitioning problems in that it must meet the underlying topology constraints, such as the

linear arrangement of physical cells in highway cellular systems and the hexagonal mesh arrangement of physical cells in cellular systems.

For highway cellular systems, we design an efficient optimal partitioning algorithm of $O(mn^2)$ by dynamic programming, where m is the number of clusters in a partition and n is the number of base stations. For hexagonal cellular systems, we develop several heuristics for multiway partitioning, based on the techniques of moving or interchanging boundary nodes between adjacent clusters. These heuristics produce optimal partitions with respect to the initial partitions obtained randomly or by centering. The heuristics are compared and shown to behave quite well through experimental testing and analysis.

Once an optimal partition of disjoint clusters is obtained, we can deploy the virtual cell system according to the topology of the optimal partition such that a cluster corresponds to a virtual cell. To analyze the performance of a virtual cell system, we adopt an open multiple class queueing network model. In addition to mobility and communication patterns among physical cells, the topology of the virtual cell system is used to determine service transition probabilities of the queueing network model. By solving the traffic equations of the queueing network model, we obtain various performance measures such as the network response time for each type of message and the utilization of the base station networks and the backbone network of the virtual cell system.

CHAPTER 1
INTRODUCTION

1.1 Objectives

The virtual cell system supports host mobility at the data link layer using high-speed base station networks for the transmission of IP datagrams in mobile computer communications. Integrating host mobility into the IP layer in an internet has been a common approach for the transmission of IP datagrams in TCP/IP environments. No existing mobile computer communications networks support host mobility at the data link layer and use high-speed base station networks.

Supporting host mobility at the IP layer and using an internet for base station networking may cause several problems in terms of the compatibility with the conventional IP protocol. Other issues that need to be addressed are the efficient location tracking strategies for mobile hosts and optimal network management and control. The objectives of this research are

- to design the communication architecture and protocol of the virtual cell system to support IP-level mobile communications,
- to optimally deploy the virtual cell system according to mobility and communication patterns among physical cells in both highway cellular systems and hexagonal cellular systems, and
- to evaluate the performance of the virtual cell system in hierarchical networking environments.

1.2 Motivations

A fixed host in TCP/IP environments is always assigned an IP address. The IP address is not only used as a unique identifier by higher layer protocols but also represents the current location of the host. An inherent problem from the transmission of IP datagrams in mobile computer communications is that the IP address of a migrating mobile host (MH) is only valid as its identification information but not able to represent its current location information. To solve this problem, various researches and developments have focused on how to integrate the functionality of host mobility into the IP layer, preserving the compatibility with the conventional IP protocol [1, 2, 3, 4, 5, 6].

Although existing mobile host protocols vary on how to represent and maintain location information for efficient tracking of MHs, the techniques of using the IP options and IP encapsulation have been considered. Typical examples of the first technique are Virtual Internet Protocol (VIP) [2, 3] and the mobile host protocol using the IP Loose Source Routing option [4]. IP-within-IP (IPIP) [1] and Internet Packet Forwarding Protocol [6] use the second technique. Even though the details of those mobile host protocols are different, we consider one representative mobile host protocol for each technique, VIP and IPIP, to illustrate common features and problems of integrating host mobility into the IP layer.

In VIP, the network layer is divided into two layers; the VIP layer resides on top of the normal IP layer. The VIP packet header is implemented as an option of the IP packet header. An MH keeps its permanent IP address used at the VIP layer for its identification and acquires a transient IP address used at the IP layer for its current location information when it migrates. If the MH is in its native network, both addresses are the same. After acquiring a transient IP address, the

migrating MH sends its native network a notification packet whose header contains the permanent and transient IP addresses. As the packet propagates to the native network, every network entity including MHs, mobile support gateways (MSGs), and even intermediate gateways on the path snoops the header information and stores address conversion information to a cache. In the same way, the header information of all data packets in transit is also used by the network entities to maintain their caches. If the source has the cache entry for the destination, the source executes address conversion to insert the correct location address in the header. The existing routing protocols of the IP layer can then correctly deliver this packet to the destination. Otherwise, the source assumes that the destination resides in its native network and sends the packet accordingly. As the packet traverses to the native network of the destination, if an intermediate gateway has the cache entry for the destination, the gateway executes address conversion and forwards the packet to the current location of the destination.

Unlike VIP, intermediate gateways in IPIP are not involved in the support of host mobility but merely in transport service. The source MSG encapsulates a network control packet for mobility management or an IP datagram from an MH into another IP datagram whose source and destination addresses specify the communicating MSGs and transmits it over an internet. The existing routing protocols of the IP layer then correctly deliver it to the destination MSG. Thus, MSGs consider the internet as a full mesh of logical point-to-point links to interconnect them. A mobile network consists of a number of MSGs, each of which maintains the location information of its own MHs. Every MH in the mobile network is assigned a unique IP address, but the network part of the IP address is the same. When the MH migrates, a forwarding pointer is set from the previous MSG to the new MSG for location tracking. If the MSG has no location information for a specific MH, it broadcasts

the other MSGs in the same mobile network an inquiry message asking who has the MH. However, IPIP also requires a transient IP address when the MH visits a foreign mobile network.

Regardless of which technique is used, the integration of host mobility into the IP layer reveals several implications. First, underlying networks differ widely in their network size, bandwidth, protocol, and packet size so that they or some of them may not meet performance needs for the support of host mobility such as rapid migration and tracking of MHs. Moreover, because they are usually under different administrative controls, efficient network management and optimization for the global mobile network may not be easy tasks.

Second, intermediate gateways may cause some performance problems no matter if they are involved in the support of host mobility or not. If they are involved, as in VIP, they must be modified or replaced to understand VIP so that the benefit from using an existing internet as a transport network is diminished. Furthermore, because intermediate gateways have to snoop every packet in transit to maintain location information and examine every data packet in transit in order to perform address conversion, the protocol processing load at intermediate gateways may severely affect overall performance. Even in those cases in which they are not involved, as in IPIP, because they usually implement packet switching operations in software, protocol processing time coupled with possible network delay between multiple hops of IP gateways may greatly affect message delivery latency.

Third, each physical cell administered by an MSG is in principle assigned an IP network address because every MH and intermediate gateway is a network entity in TCP/IP environments. It means that an MH migrating to a different physical cell requires a different network address to represent its current location. Some undesirable features of IP-level mobile host protocols essentially come from this fact.

VIP requires a large amount of the transient IP address space, which becomes a very scarce network resource because internets are rapidly growing. IPIP uses one permanent IP address for each MH but relies on the broadcast inquiry among MSGs when the location information is not available. This restricts the scalability of IPIP to a local area.

Fourth, although IP-level mobile host protocols based on options keep the compatibility with the conventional IP protocol, in a practical sense they may interfere with it because most existing fixed hosts and intermediate gateways do not implement the IP options. The implementations to support the IP options may not be feasible in the near future. In addition, current efforts to provide IP multicasting protocols and connection-oriented protocols require IP-level mobile host protocols to be compatible.

As high-speed, connectionless, packet-switched networks are emerging to extend LAN-like performance across a wide area, we believe that they can greatly affect the support of host mobility in mobile computer communications. Examples are ATM networks [7, 8] and Switched Multimegabit Data Services (SMDS) networks [9, 10], which are subnetworks providing a MAC service across a wide area in a large interconnected network. To take advantage of these networks, it is desirable to establish the support for host mobility at the data link layer between base stations (BSs).

1.3 Design of the Virtual Cell System

In this research, we design the network infrastructure of the virtual cell system for the transmission of IP datagrams in mobile computer communications. The virtual cell system shields host mobility from the IP layer by supporting it at the data link

layer using base station networking. To design the virtual cell system, there are several issues to be addressed:

- analyzing the requirements of base station networks,
- identifying the current location of MHs without using their IP addresses,
- constructing a distributed location information structure for efficient location tracking,
- reflecting the impact of host mobility on the distributed location information in the design of protocols,
- designing the communication architecture and protocol, and
- harmonizing the communication architecture and protocol with the conventional IP protocol.

Considering the requirements of base station networks from two different viewpoints, application and mobility management, the virtual cell system takes advantage of high-speed datagram packet-switched networks with the multicast ability for base station networking. The base station networks interconnect remote bridge BSs, so as to preserve the interconnection level of physical cells at the data link layer. The current location of MHs is identified by the base station network address. It represents which BS can communicate with a particular MH. For efficient location tracking of MHs, the virtual cell system constructs a distributed hierarchical location information structure. Based on the distributed hierarchical location information, the virtual cell protocol for handoff, address resolution, and data transfer is designed. The handoff procedure can utilize the multicast ability of the base station network to achieve the consistency of the distributed location information. Since the IP network address of

a migrating MH represents at least the near location information of the MH in the virtual cell system, the distributed location information coupled with the existing routing protocols of the IP layer give the same effect of maintaining a conceptually centralized server for the whole mobile network.

1.4 Optimal Deployment of the Virtual Cell System

A virtual cell should accommodate a limited number of BSs in order to produce an acceptable performance no matter how the network parameters such as the topology and capacity of the virtual cell system are engineered. In addition, the cost of tracking MHs within a virtual cell is lower than between virtual cells. Thus, in order to decrease the total communication cost of tracking MHs, it is intuitively desirable that the two BSs between which the MH migrates or communicates very often should be contained in the same virtual cell, while the two BSs between which the MH migrates or communicates infrequently could be separated into different virtual cells.

The location tracking of MHs involves two primitive operations in the virtual cell: the *move* operation to perform handoff and the *find* operation to locate the current BS of an MH for the transmission of IP datagrams. Then mobility and data communication patterns among BSs can be represented by the frequencies of move and find operations among BSs. Even though there is a tradeoff between the costs of move and find operations within a virtual cell, the optimal deployment of the virtual cell system is concerned with optimal partitioning of BSs into virtual cells, so as to minimize the total communication cost for a sequence of move and find operations in the entire system.

In addition to mobility and communication patterns among BSs, the optimal deployment of the virtual cell system also considers the physical arrangement of BSs so that BSs in a virtual cell are contiguously connected in their underlying topology,

not scattered. While the frequency is represented by a complete directed graph among BSs for a possible communication between MHs through different BSs, the physical topology is represented by a linear graph in highway cellular systems or by a hexagonal mesh graph in cellular systems. Hence, the optimal deployment of the virtual cell system is concerned with the two types of graphs, the topology graph for partitioning BSs and the frequency graph for optimizing communication cost.

For general graphs, the problem of finding a cover of disjoint sets such that the sum of all edge weights, whose two endpoints are in two different sets, is equal to or less than a given positive integer is known as NP-complete for the arbitrary size of a set [13]. The k-cut problem of finding a partition of vertices into k nonempty clusters such that the total edge weight between clusters is minimum is also NP-complete for arbitrary k. The k-cut problem with specified vertices, which is an extension of the 2-cut problem solvable via repeated applications of a max-flow min-cut algorithm, becomes NP-hard even for $k = 3$ [14]. Given a planar graph with an s-outerplanar embedding, an optimal solution for maximum independent set can be obtained in time $O(8^s n)$ by a dynamic programming technique, where n is the number of nodes [15]. If an n-vertex planar graph G is given with an s-outerplane-separable planar embedding of G, then the optimal partition of two clusters of fixed size is determined in time $O(s^2 n^3 2^{3s})$ by a dynamic programming technique [16]. This implies that without considering the frequency graph, the 2-way partition in a hexagonal mesh of cellular systems is exponential where $s = \Theta(n^{1/2})$.

Given the frequencies of move and find operations among n BSs in highway cellular systems, we develop an efficient optimal partitioning algorithm of $O(mn^2)$ for an arbitrary number of clusters m by dynamic programming. In a hexagonal mesh of physical cells, we develop several heuristics for multiway partitioning, based on the techniques of moving or interchanging the boundary nodes between adjacent clusters.

These heuristics produce optimal partitions with respect to the initial partition obtained randomly or by centering. The heuristics are compared and shown to behave quite well through experimental tests and analysis.

1.5 Performance Analysis of the Virtual Cell System

Once an optimal partition of m disjoint clusters is obtained, the virtual cell system can be deployed so that each cluster corresponds to a virtual cell. Virtual cell i, where $1 \leq i \leq m$, is constructed by interconnecting the BSs of the ith cluster and a location server by a base station network. These m virtual cells are interconnected by the backbone network.

To evaluate the performance of virtual cells, we apply an open multiple class queueing network model. There are three types of messages entering or leaving the virtual cell system via BSs: the handoff message, the data message, and the address resolution message. The handoff messages are generated due to move operations, and the data and address resolution messages are due to find operations. The move and find frequencies in conjunction with the topology matrix of the virtual cell system obtained by the optimal algorithms are used to determine service transition probabilities for each type of message in the queueing network model.

With various system parameters, we can conduct interesting sensitivity analyses to determine network design tradeoffs. The first application of the proposed model is to determine an adequate network bandwidth for base station networks such that the networks would not become a bottleneck. Also, we can evaluate the network utilization due to various messages. For instance, when the vehicles begin moving fast, the migration rate will be increased. This implies more handoff messages and more forwarding operations. The network traffic should be increased accordingly.

The performance analysis results should provide a good evidence to demonstrate the system efficiency under different mobility assumptions.

CHAPTER 2
SURVEY OF RELEVANT WORK

In this chapter, we survey three representative approaches in mobile data communications: Cellular Packet Switch (CPS) [11, 12], IP-within-IP (IPIP), and Virtual Internet Protocol (VIP). All three approaches commonly take advantage of packet-switching technology to keep track of mobile users. However, CPS is designed for digitized voice and data services in personal communications and handles virtual circuit connection at the data link layer for mobility management. IPIP and VIP are designed for TCP/IP environments and integrate host mobility into the IP layer.

2.1 Cellular Packet Switch

The increasing demand in wireless communications has propelled the development of new and improved cellular and cordless systems. Even with these systems, however, increased voice traffic together with the ever growing demand for data services have revealed capacity problems. To provide the necessary capacity in a densely populated area, more efficient use of the limited radio spectrum will be required, and a closely spaced grid of microcells will have to be deployed. While small cells mitigate capacity problems, the reduced cell size makes the task of locating and controlling the mobile users substantially more complex. CPS is intended to support high density personal communication networks using a new network architecture based on packet switching. A packet switched architecture itself enables distributed control of mobility management and radio resource management functions because the addresses and other information in packet headers make it possible for dispersed network elements

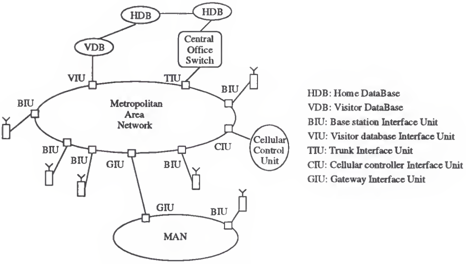


Figure 2.1. Cellular Packet Switch Architecture

to respond to user mobility without the intervention of central controllers. In addition, it is efficient in integrating the diverse types of information that will be offered to future networks.

2.1.1 Network Architecture

As shown in Figure 2.1, CPS takes advantage of a Metropolitan Area Network (MAN) based on the IEEE 802.6 standard in order to support the increased traffic resulting from the use of microcells. The MAN connects base stations, control units, databases, and links to fixed information services, such as the Public Switched Telephone Network. In addition to routing, the interface units are responsible for information transfer and protocol conversion through different media in the network fabric including the MAN, the fixed network, and the radio transmission systems. In CPS, call processing (setup and release) is handled by the central office switch, while mobility management and radio resource management are handled on the MAN. Information on the MAN is organized in data units compatible with ATM, and data

units are routed to their correct destinations according to addresses in packet headers. The base station interface units translate information between the MAN format and that of the radio link. The trunk interface units convert user information between the MAN protocol and the fixed network protocol, which may be ISDN or broadband ISDN, for example. The home database interface unit could connect the MAN directly to a database that serves CPS. It also could provide connectivity to a general purpose data network connected to home databases. Other interface units, such as the cellular controller interface unit and the visitor database interface unit, simply provide their network elements with access to the MAN. Because the coverage area of a single MAN could be as low as a few square kilometers in densely populated areas, the gateway interface unit interconnects adjacent MANs. It permits CPS to perform handoffs between base stations served by different MANs.

2.1.2 CPS Protocols

CPS protocols distribute network control functions among the interface units using virtual circuit packet switching. The link between a wireless terminal and the fixed network is a virtual circuit identifier (VCI), which is a number carried in the header of each packet on the MAN. By referring to this VCI, MAN interface units route each packet to its correct destination. Thus, this VCI establishes a logical link between interface units. At the beginning of a call, the cellular controller interface unit assigns a call-specific VCI to the call. The base station interface unit that initially handles the call and the trunk interface unit that moves the call to and from the fixed network both record the VCI. When the call moves to a new base station, that base station records the VCI. It also sends a message to the previous base station telling that base station to delete the VCI from its table.

The following example shows a wireless terminal receiving information from a trunk interface unit when it moves from the service area of base station BS_1 to that of base station BS_2. Initially, the terminal is communicating with BS_1 and has a call assigned VCI 100 by the cellular controller interface unit. First the terminal determines that the radio link to BS_1 is inadequate and that an acceptable link is available to BS_2. It spontaneously sends a handoff message to BS_2. This message contains VCI 100 and the identity of BS_1. BS_2 enters VCI 100 in its address table and sends a message to BS_1, which causes BS_1 to delete VCI 100 from its table. The handoff message from the terminal to BS_2 contains the sequence number of the last nonvoice packet received accurately through BS_1. It relays this information to the trunk interface unit, which transmits to BS_2 nonvoice packets that it previously sent to BS_1 while the handoff was in progress. Thus, all further transmissions from the trunk interface unit go through BS_2.

The handoff is terminal initiated, which means that the terminal makes the decision that a handoff is necessary. It is also terminal controlled, which means that the terminal decides to move the call to another base station. In the overlapped area between two different radio cells, the terminal communicates with either base station, but not with both simultaneously. The network stores, for possible retransmission, critical information (data packets) in a source buffer at the trunk interface unit. It could also store this information in a destination buffer in the base interface unit.

In addition to this handoff procedure, CPS supports other protocols that perform a variety of call processing and mobility management functions, including terminal initiated call setup, network initiated call setup, terminal initiated call release, network initiated call release, location updating and authentification, inter-MAN handoff, and path optimization. At the wireless terminal and in the fixed network, the call setup and release procedures are the Q.931 protocols of ISDN. Within CPS these

protocols are augmented by additional functions necessary to enter or delete a VCI from the address tables in the WIU, the initial base station interface unit, and the trunk interface unit. For location tracking, protocols developed for existing cellular systems are used. Meier-Hellstern et al. [12] describes wireless services based on the IEEE 802.6 standards in detail.

2.2 Virtual Internet Protocol

Teraoka, et al. [2, 3] identify the need of host migration transparency to provide the same computational environment with mobile users in a large interconnected network. The main goal is to support host portability, which implies no communications during the migration of mobile hosts. To achieve host migration transparency, the virtual network concept that uses the propagating cache method has been proposed. As an example of the virtual network, VIP is derived from the conventional IP protocol.

2.2.1 Network Architecture

In order to provide a transport layer with host migration transparency, two network layer identifiers of a host are introduced: one is migration independent and the other is migration dependent. The transport layer specifies the target host by the migration independent identifier so that the transport layer is not aware of host migration. To support this migration independent identifier, the concept of virtual network is introduced. The virtual network is a logical network. Each host is considered to be permanently connected to a particular virtual network, called the *home network* of the host. A host never migrates among virtual networks even if it migrates among the physical networks.

From the prospective of the network protocol architecture, a conventional network layer is divided into two sublayers: *virtual network sublayer* (VN-sublayer) and

physical network sublayer (PN-sublayer). A *virtual network address* (VN-address) is assigned to a host in the VN-sublayer, and a *physical network address* (PN-address) is assigned to a host in the PN-sublayer. The VN-address of a host never changes no matter where the host migrates, while the PN-address changes if the host migrates. Thus, the VN-address of a host indicates its home network, to which it was initially connected.

2.2.2 Propagating Cache Method

Since the transport layer specifies the target host by its VN-address, the main function of the VN-sublayer is to convert a VN-address into its corresponding PN-address. To solve this problem, the propagating cache method is proposed. In the propagating cache method, each host and gateway has a cache for address conversion, called an Address Mapping Table (AMT). If the source host has an AMT entry for the destination host, the source host executes address conversion before sending out a packet; the PN-sublayer can then correctly deliver this packet to the destination host. Otherwise, the source host assumes that the destination host is connected to its home network and sends the packet accordingly. As the packet traverses the network to the home network of the destination host, if an intermediate gateway has the AMT entry for the destination host, the gateway executes the address conversion and forwards the packet to the current physical location of the destination host. Whenever a host or gateway receives a packet, the VN-sublayer creates or updates the AMT entry for the source host of the received packet. Thus, the AMT information propagates across the interconnected networks as communication progresses.

2.2.3 Communication Procedures

VIP is derived by applying the virtual network concept to the IP layer. The conventional IP layer is divided into two sublayers: the VIP sublayer and the IP

sublayer. A host has an IP address in the IP sublayer and a VIP address in the VIP sublayer. Since the VIP address of a host never changes even if the host migrates to another network, the TCP/UDP layer and its application programs can uniquely specify a host by the VIP address. The IP address becomes invisible to the TCP/UDP layer. Since the IP address is location dependent, however, it changes when a host migrates to another network. This means that the host must be assigned a new IP address. The function to allocate an IP address to a host is not included in VIP and implemented with daemon processes.

In packet transmission, the TCP/UDP layer issues a transmission request to the VIP sublayer with the VIP address of the destination host. The VIP sublayer then generates the VIP header and sets each field. Next, the VIP sublayer searches for the AMT entry for the destination host. An AMT entry consists of four data fields: *VIP address* used as key, *IP address* of the host, *address timestamp* to specify the version number of the VIP/IP address pair of this entry, and *idle timer* used for aging the AMT entry. If the entry is found, the VIP sublayer converts the VIP address of the destination host into the corresponding IP address. Otherwise, the VIP sublayer assumes that the VIP address and the IP address of the destination host are the same. This means that the VIP sublayer assumes that the destination host exists in its home network. The procedures in the IP sublayer are the same as that of the conventional IP layer.

In packet reception, some modifications to the existing IP procedures are necessary. When the data link layer notifies the IP sublayer of packet reception, the IP sublayer calls a function in the VIP sublayer that creates or updates the AMT entry for the source host of the received packet. If the AMT entry for the source host does not exist, an entry is created. If the entry already exists and either the IP address of the source host has changed or the value of the timestamp field of the received packet

is newer than that of the timestamp field in the AMT entry for the source host, then the entry is updated.

After this modification to the AMT entry, the IP sublayer compares the destination IP address field of the received packet with its own IP address. If the two IP addresses match, the IP sublayer passes the packet to the VIP sublayer. Otherwise, the IP sublayer calls another function in the VIP sublayer in order to convert the VIP address of the destination host into the corresponding IP address, if necessary. If the AMT entry for the destination host is found and the value of the timestamp field of the received packet is older than that of the timestamp field in the AMT entry for the destination host, this function returns the values of the IP address field and the timestamp field. If the address conversion occurs, the IP sublayer modifies the destination IP address field in the IP header and the VIP sublayer modifies the timestamp field in the VIP header. The packet is then forwarded to the next gateway or to the destination host.

2.2.4 Migration Procedures

When a migrating host is connected to a subnetwork, the host sends a connection notification packet to its home network and waits for a connection acknowledgement packet. The connection notification packet announces that the source host has just been attached to a subnetwork and is then relayed by gateways. Since this connection notification packet includes the VIP and IP addresses of the source host, intermediate gateways can learn the relation of these two addresses and create or update the AMT entry for the source host. The home gateway, which is the gateway in the home network of the migrating host, receives the connection notification packet, creates the AMT entry for the migrating host, and returns the connection acknowledgement packet to it. Note that the home gateway never deletes this AMT entry by timeout.

The home gateway also broadcasts the connection notification packet within the home network of the migrating host if the home network is a broadcast type network such as Ethernet. In addition, the home gateway creates an entry in the ARP table for the migrating host to answer the ARP request for that host.

Upon packet transmission to a migrating host, if the source host does not have the AMT entry for the migrating host, the host assumes that the IP address of the migrating host is equal to its VIP address. The packet then heads for the home network of the migrating host. If a gateway on the path has the AMT entry for the migrating host, the gateway converts the VIP address into the corresponding IP address and forwards the packet to the actual location of the migrating host. If the migrating host returns a packet to the source host, the source host can create the AMT entry for the migrating host so that the host can thereafter directly specify the IP address of the migrating host.

When a migrating host is about to disconnect from a subnetwork, it sends a disconnection notification packet to its home gateway. When the home gateway receives that packet, it broadcasts a control packet to all subnetworks to which it is connected so that a relevant AMT entry is deleted. Gateways in the subnetwork receive the control packet. If a gateway has the AMT entry for the migrating host, it deletes the entry and also broadcasts the control packet to any other connected subnetworks. If a gateway does not have an AMT entry for the migrating host, the gateway does nothing. Thus, most AMT entries of the migrating host are deleted when the host is about to disconnect from a subnetwork.

2.3 IP-Within-IP

IPIP is designed to provide continuous network access to mobile hosts in a campus environment. The key idea is to assign Mobile Support Stations (MSSs) to every radio

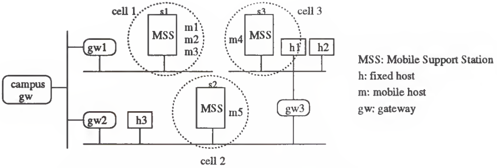


Figure 2.2. A Model for Typical Campus Environments

cell on which mobile internetworking protocols are supported, and attach them to an interconnected network. Figure 2.2 shows a typical campus environment used in this work.

2.3.1 Network Architecture

All MHs have the same internet network address so that they can be immediately distinguished from fixed hosts by their IP addresses. In addition, an MH has a unique IP address regardless of its migration. This eliminates the need of a transient IP address, the change of the IP address at name servers, and the notification of it to higher protocols. Thus, MSSs are responsible for the addressability of all MHs on the network. An MSS has its regular IP address on the wired network, a mobile IP address for its wireless interface, and a second mobile IP address for its wired interface. The radio cell is the geographical area around the MSS and is defined by the propagation characteristics of the electromagnetic waves at the operating frequency.

As a requirement, link-level broadcasts are assumed in order for the MSS's beacon to be received from all the MHs in a radio cell. The ability of the hardware to detect signal strength and report it to the device driver is also required to realize that the MH is at the outer reaches of a radio cell and that it should try to switch to a different one. In addition, in the case of spread-spectrum or multi-channel radio

communications, the ability to receive on multiple channels at once is assumed so as to detect areas where radio cells overlap and decide when MHs switch to a new MSS whose signal is stronger than another MSSs.

2.3.2 Data Link Layer

There are two kinds of interactions that take place in the data-link layer: mapping logical addresses to physical addresses and dynamically acquiring IP addresses when moving to a foreign campus. For address mapping, we have the following problem in mobile networking. When an MH wants to communicate with another MH, and the two MHs have IP addresses in the same subnet, the first MH will send an ARP request for the physical address of the second, since it thinks the latter is on the same connected network. If the second MH is in the same radio cell, then it will respond to the ARP packet. If it is not, the MSS will execute the proxy ARP protocol for that other MH. When that happens, the first MH will send all traffic destined for that other MH through its MSS, which will now encapsulate it in an IPIP datagram, and deliver it to the remote MSS handling the second MH's traffic.

When an MH moves into a foreign campus, it requests a transient IP address, which it will use to talk to the local MSS and also to encapsulate IPIP packets in order to maintain open connections to its home campus. The MH knows it moved into a foreign campus if the network portion of the MSS's IP address is different from its own. This can be achieved by extending a number of protocols such as the Reverse Address Resolution Protocol and the Bootstrap Protocol that provide such physical-to-logical address mappings as part of their functionality or by using existing and under-development protocols such as the Dynamic Host Configuration Protocol.

2.3.3 Network Layer

The IP layer of the MSS must be modified to decide how to route a datagram. An entry is kept in the kernel for each MH the MSS knows about. The key field in each entry is the MH's home address. Additional fields include, for local MHs, the IP address to be used when communicating the MH if it is not the same as the home address; for remote MHs, the IP address of the MSS handling the MH's traffic; a timer field, which is reset every time the entry is accessed and is used to expire entries that have not been used recently. There are two IP-level protocols: *IP-within-IP* (IPIP) used for the encapsulation and *Mobile Internetworking Control Protocol* (MICP) used to locate the other MSS.

The IPIP protocol is used to tunnel IP datagrams from one cell to another or from one network to another, essentially using IP for virtual point-to-point connections. The problem is how to deliver an IP datagram to an MH whose location cannot be deduced from its IP address. MSSs keep track of MHs and tunnel packets from MSS to MSS using IPIP encapsulation, which is equivalent to IP loose source routing. However, they differ in that source routing depends on an IP option being implemented on all routers along a path, whereas encapsulation only depends on a protocol module being implemented at the two endpoints of the tunnel.

When an IP datagram is to be sent out at the IP layer, the routing and MH tables are consulted. If the destination MH is in another cell with a known MSS, it is encapsulated in an IP datagram and sent to the remote MSS. Thus, the IP destination and source addresses of the datagram can be considered the two endpoints of the tunnel. Upon receipt by the remote MSS, the IP code hands the datagram to the IPIP protocol module. The latter strips the IPIP header and, after decrementing the time-to-live field, feeds the packet, which now has the real destination and source IP

addresses, back into the IP queue so that it can be delivered. In addition, the code checks whether the source of the IPIP packet was another MSS.

The MICP is used by the MSSs to exchange control information and by the MHs to signal to their MSSs that they have changed cells. MICP datagrams are also encapsulated in IP datagrams with a different protocol type number. There are three classes of MICP traffic: MH-MSS handshakes, MSS-MSS exchanges to distribute MH location information, and MSS configuration exchanges.

The MSS periodically broadcasts its identity. An MH may receive beacons from more than one MSS at a time. This is usually the case where adjacent cells of radio networks overlap. If the hardware gives an indication of the signal strength, and the MH is within the overlapping area of two cells, the MH may want to switch to the cell that has the strongest signal. Otherwise, the MH may wait until it no longer receives the beacon from the previous MSS before it switches to the new one.

The beacon is an MICP packet that contains the IP address of the MSS's primary interface and the subnet mask of the current cell. It is broadcast periodically on the local broadcast address of the cell. When an MH receives a beacon packet that is different from the previous one it received, it responds with a greeting MICP packet. This packet contains the home address of the MH, the IP address of the MSS of the previous cell it was in.

When the MSS receives a greeting packet, it responds with a greeting acknowledgement MICP packet. It updates the relevant system structures to indicate that the MH is now local and records the information supplied by the MH. Upon receipt of the greeting acknowledgement packet, the MH changes its routing tables to route all packets through the new MSS and now considers it its new MSS.

Beacon packets should be broadcast often enough that missing some will not cause the MH to assume it is no longer in the cell. If the greeting packet is lost, the MSS will

not respond to it, and the next beacon packet will cause the MH to send a new greeting. If the acknowledgement packet is lost, the MH will send another greeting until it gets an acknowledgement. After the first successful beacon/greeting/acknowledge handshake, duplicate beacons are ignored. Duplicate greetings cause the MSS to overwrite the MH information, and duplicate acknowledgements have no effect since the MH can tell they are coming from its current cell.

When an MSS receives the greeting, it also sends a forwarding pointer MICP packet to the MSS previously handling that MH. The previous MSS uses this information to handle properly any traffic for that MH originating in the segments of the network it is on. The MSS will keep retransmitting the forwarding pointer until it gets a forwarding acknowledgement MICP packet. This is needed in order to make certain that the previous MSS will not attempt to handle traffic for an MH that it no longer serves. After an MH moves away from a cell, its MSS will continue receiving IPIP-encapsulated datagrams for it from other MSSs. It knows where the MH migrated, so it informs those MSSs of the fact by sending them a redirection MICP packet.

When an MSS receives from another MSS a datagram for an MH that it no longer serves, but for which it has in its cache the address of the MSS currently serving it, it will send a redirection packet to the sender. To avoid having to rely on timeouts from higher-level protocols, the MSS will also attempt to deliver the received datagram to where it thinks it should have gone. If the MH had moved again, this delivery will result in a new redirect being sent. Eventually, all MSSs involved in a particular MH's traffic will either converge to point to the same MSS, or their information will expire.

Assuming that the MH does not change cells faster than these packets can travel, the network should be able to track the MH at all times. The only danger with this

redirect packet being lost is that the originating MSS will not be informed of the MH's movement. This does not cause problems, since if another packet is received, another redirect will be sent. Eventually, one will reach the offending MSS. Hence, no explicit acknowledgement of the redirect packet is needed.

When an MSS is asked to deliver a packet to an MH it does not know about, it must discover which MSS is responsible for it. It drops the packet and then sends an MICP packet to all the MSSs on the campus to find who is responsible for it. The MSS currently handling the MH will respond with an MICP packet. This on-demand discovery of peer MSSs allows them to know only the MSSs involved in their MHs' traffic.

2.4 Chapter Summary

The existing mobile data networks can be largely categorized into two different groups depending on what applications are intended: TCP/IP applications and packetized voice and data services. In the mobile data networks such as IPIP and VIP for TCP/IP applications, a common approach is to implement the IP-level mobile host protocols and use the internet for base station networking. On the other hand, in the mobile data networks such as CPS for personal communications, the importance of base station networking has been emphasized and MAN technologies are used for distributed mobility management at the data link layer. However, it has not considered TCP/IP environments, which accommodate a large population of the internet users and applications.

The next chapter is about the virtual cell approach to solve the problems from integrating host mobility into the IP layer and using an internet for base station networking in TCP/IP environments.

CHAPTER 3
THE VIRTUAL CELL SYSTEM

This chapter describes a *virtual cell* concept for the transmission of IP datagrams in mobile computer communications. A virtual cell consists of a group of physical cells whose base stations are implemented by remote bridges and interconnected via high-speed datagram packet-switched networks, supporting host mobility at the data link layer. Thus, as far as the IP layer is concerned, it appears as if the communication between two mobile hosts in different physical cells were taking place directly as in the same physical cell. It eliminates the necessity of IP-level protocols for mobility management, which may interfere with the conventional IP protocol in a practical sense, and achieves a logically flexible coverage area according to mobility and data traffic patterns among physical cells. Virtual Cell Protocol (VCP), which consists of handoff, address resolution, and data transfer modules, is designed based on the distributed hierarchical location information of mobile hosts.

3.1 Virtual Cell Concept

Consider the transmission of IP datagrams between two MHs within a physical cell. Because radio links provide the broadcast ability, the source can deliver IP datagrams to the destination using the normal Address Resolution Protocol (ARP). Thus, the broadcast ability of radio links eliminates the necessity of locating MHs, resulting in no mobility management protocols at the IP layer and no modifications to the normal ARP protocol. To apply a similar rationale to MHs crossing the physical cell boundary, we propose the virtual cell concept, as depicted in Figure 3.1.

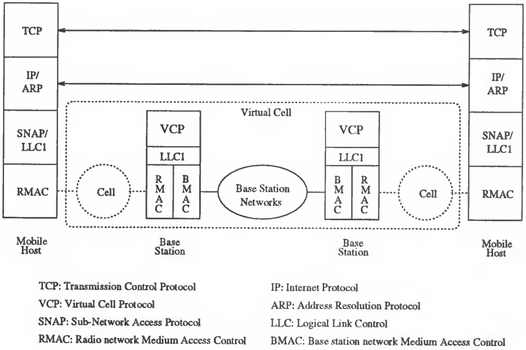


Figure 3.1. Virtual Cell Concept

A virtual cell is a logically extended cell of physical cells whose BSs are implemented by remote bridges and interconnected via a base station network. Because remote bridge BSs make it possible to preserve the interconnection level of physical cells at the data link layer, the whole virtual cell is assigned an IP network address. In order to make it possible for MHs in a virtual cell to directly deliver IP datagrams using the conventional IP and ARP protocols, a mobility management protocol, called Virtual Cell Protocol (VCP), is implemented at the data link layer of BSs. Each virtual cell is also allocated an ARP/Location server (ALS), which implements VCP and supports the normal gateway function with other virtual cells or fixed networks. VCP supports handoff, address resolution, and location tracking for datagram delivery, based on the distributed hierarchical location information of BSs and the ALS, as described in detail in Section 3.3.

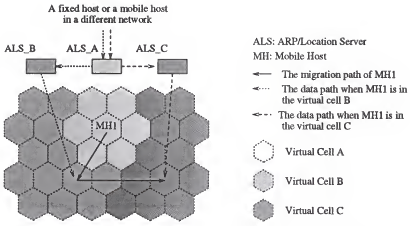


Figure 3.2. Data Path Between Virtual Cells

It should be noted that the identification and location information of an MH is represented by two different data link layer addresses in the virtual cell, Radio network MAC (RMAC) address and Base station network MAC (BMAC) address, respectively. Note that the BMAC address is not only used as the location information of the MH but also as the identification information of a virtual cell. For example, the SMDS addresses can be formatted with a similar structure used for the North American Numbering Plan to represent the geographic semantics. This separation eliminates the necessity of acquiring a temporary address to represent the MH's current location as it migrates. In addition, unlike IP-level mobile host protocols, the IP network address also represents the correct location information of the MH while it moves to a different physical cell in the virtual cell. This is possible because the whole virtual cell is assigned an IP network address and host mobility is shielded from the IP layer.

Even when the MH migrates between virtual cells, the IP network address is still valid as the near-location information in the virtual cell environment. Figure 3.2 shows the data path from a fixed host or an MH in a different network to MH_1, which migrates from its native virtual cell A to virtual cell B to virtual cell C.

Whenever MH_1 crosses the boundary of virtual cells, a one-hop forwarding pointer is maintained from ALS_A in its native virtual cell to the ALS of its current virtual cell, ALS_B or ALS_C, by the handoff procedure of VCP. Note that the forwarding pointer is not maintained at the IP layer but at the VCP layer. When a fixed host or an MH in a different network wants to send an IP datagram to MH_1 in virtual cell B or C, the existing routing protocols of the IP layer correctly deliver it to ALS_A. Then ALS_A redirects it at the VCP layer to the corresponding ALS, ALS_B or ALS_C, which keeps track of the exact location of MH_1 in its virtual cell.

Thus, every MH resides in its native virtual cell from the IP layer's viewpoint but practically may reside in an adjacent virtual cell because of a one-hop forwarding pointer from the native virtual cell to the current virtual cell maintained at the VCP layer. It means that the combination of the existing routing protocols of the IP layer and the forwarding pointer between virtual cells at the VCP layer gives the same effect: that we have a fully duplicated location information among virtual cells for the global mobile network.

To configure a number of neighboring physical cells into virtual cells, the underlying transport network should provide high-speed datagram packet-switched services and the group addressing capability at the data link layer. The flexible configuration allows network managers and designers to customize the logical coverage area of the virtual cell to their requirements according to host mobility and communication patterns among physical cells.

Consider an environment in which a number of physical cells are deployed in a metropolitan area. The cell size in the urban area is usually smaller than that in the suburban area in order to accommodate a larger population of mobile users. Even though user mobility is inherently unpredictable, there could be a high possibility that the daily routine of mobile users is usually confined to several physical cells in

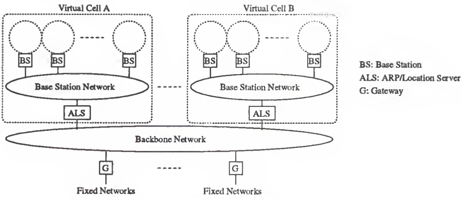


Figure 3.3. Virtual Cell Architecture

the urban area. Because of a relatively small cell and large population of mobile users in the urban area, it becomes very important to achieve the fast location tracking of mobile users and the small cell switching latency between physical cells, mitigating the effect of a large volume of mobility management information. Thus, the physical cells covering the urban area could be combined into one or a few number of virtual cell(s) for efficient mobility management and data delivery.

3.2 Virtual Cell Architecture

Because the virtual cell is a logical concept, the same transport network may support several virtual cells simultaneously. As shown in Figure 3.3, there are two roles of the transport network: base station network and backbone network. The base station network is used to interconnect a number of BSs and an ALS to build a virtual cell, and the backbone network is used to interconnect among virtual cells and fixed networks. Note that the base station network utilizes both point-to-point and multicast communications, while the backbone network uses only point-to-point communication.

3.2.1 Base Station Network

The base station network should meet some requirements from two different viewpoints:

- *Application's viewpoint.* IP operates under the assumption that packet arrivals are independent and unpredictable. If IP is coupled with the connection-oriented base station network, one connection establishment and release may be needed for each individual packet over a fixed link between a pair of BSs. Although a bundle of packets could be transmitted over one connection by some multiplexing techniques, when communicating MHs migrate to different physical cells with the connection, the problem becomes complex and even intractable. In addition, because location tracking has inherently connectionless properties, the base station network should support the datagram delivery. The current ubiquitous coverage of TCP/IP networks and applications also requires that the base station network be able to cover a wide area.

- *Mobility management's viewpoint.* Combined with the distributed location information of MHs among BSs, host mobility is a main cause of the inconsistency. In order to maintain the consistency of the distributed location information efficiently, the base station network should support the multicast ability. Furthermore, because the network control information for mobility management is expected to be greatly increased as the cell size becomes smaller, the base station network should have high-bandwidth not so as to affect the normal data traffic greatly.

We consider SMDS as an example with these characteristics. Both individual and group addressing capabilities combined with a set of addressing-related service

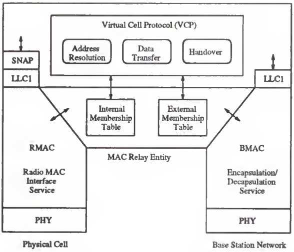


Figure 3.4. Base Station Architecture

features (e.g., source address validation, source and destination address screening) enables to create a number of logical networks over SMDS. Every BS and ALS is attached to a Subscriber Network Interface (SNI) and individually addressed. At the same time, a group address identifies all BSs in a virtual cell, ensuring that each group address identifies uniquely only one set of individual addresses. The number of SNIs to be supported by a switching system is at least 256 SNIs and the future number is up to 4096 SNIs within a Local Access Transport Area. Considering that the range of a physical cell is usually 3-5 km in the urban area or 10-15 km in the suburban area, the capacity of a very few number of switching systems may be enough to support base station networking within a metropolitan area.

3.2.2 Base Station

The communication architecture in a BS is shown in Figure 3.4. The internal port connected to a physical cell implements an RMAC entity, while the external port connected to the base station network implements a BMAC entity. The MAC relay entity translates the information format between the physical cell and the base station network. The protocol identifier field of both the RMAC and BMAC frame

headers is set for LLC type 1 Unnumbered Information format. For example, the Higher Layer Protocol Identifier field of the SMDS Interface Protocol L3_PDU used as BMAC frames is set to 1. The LLC Service Access Point of the RMAC frame is set for Subnetwork Access Protocol (SNAP), while that of the BMAC frame is set for VCP.

The VCP header has three fields: VCP type, VCP length, and RMAC address. The VCP type is set to one of *ADDRESS*, *DATA* and *HANDOVER* to specify the address resolution, data transfer and handoff modules of VCP, respectively. Depending on the VCP type, the information field of the VCP frame has a different format. The *DATA* type indicates that the information field has an IP datagram, and the information fields for the *HANDOVER* and *ADDRESS* types are given in the next section. The VCP length indicates the length of the VCP header. Note that the RMAC address is only used with the *DATA* type.

Each BS maintains a partial location information of the virtual cell in its Internal Membership Table (IMT) and External Membership Table (EMT). IMT keeps track of all MHs currently roaming in its physical cell. The IMT entry for an MH has two fields, the IP and RMAC addresses, each of which is used as an identifier but has a different role. The IP address is used as an identifier by the higher network layers as usual, but the network part of the IP address represents the MH's native virtual cell to which it initially belongs. This IP address obtained by the handoff procedure is only needed for fast address resolution, as described in the next section. On the other hand, the RMAC address is the other identifier used by VCP for the support of host mobility at the data link layer. EMT has a relatively small number of entries for MHs roaming in different physical cells of the same virtual cell. The EMT entry for an MH has three fields, the IP, RMAC and BMAC addresses, respectively, of which the BMAC address represents the MH's current network address.

If the source MH wants to send an IP packet to the destination MH whose native virtual cell is the same, the source performs ARP and sends the IP packet using the destination RMAC address. On the other hand, if the source wants to send an IP packet to the destination whose native virtual cell is different, the source does not perform ARP and sends the IP packet using the RMAC address of its current BS, which is obtained by a beacon message. When the BS receives an SNAP/LLC/RMAC frame from the internal port, it checks the Protocol Identification (PID) field of the SNAP header. If the PID field is set for ARP, the ARP packet is sent to the address resolution module which performs Virtual Cell Address Resolution Protocol (VCARP). VCARP achieves the IP-to-RMAC address binding for the destination and also distributes the location information of the source and destination. If the PID field is set for IP, the IP packet is sent to the data transfer module with the destination RMAC address. If the destination RMAC address is the BS's RMAC address, the data transfer module sets the RMAC address field of the VCP header to the BS's RMAC address and relays a VCP/LLC/BMAC frame to the ALS without searching IMT and EMT. Otherwise, the data transfer module keeps track of the destination MH's current location using IMT and EMT. If the corresponding entry is not found, the data transfer module transmits to the ALS a VCP/LLC/BMAC frame whose VCP header contains the destination RMAC address.

3.2.3 ARP/Location Server

Each ALS maintains Global Membership Table (GMT) for MHs roaming in its virtual cell, as shown in Figure 3.5. The entry format of GMT is the same as that of EMT. There are two types of frames from the backbone network for data transfer. One is the VCP/LLC/BMAC frame redirected from other virtual cells that maintain one-hop forwarding pointers at the VCP layer, and the other is the

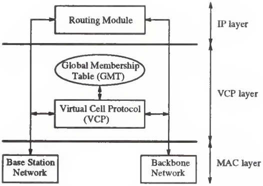


Figure 3.5. Communication Architecture in an ARP/Location Server

SNAP/LLC/BMAC frame from fixed networks or other virtual cells. The first type of frame is sent to the data transfer module which identifies the destination BS by searching GMT with the RMAC address of the VCP header, and transmits it to the base station network. The second type of frame is, however, directly sent to the routing module which extracts the IP packet. If the network part of the destination IP address indicates the same virtual cell, the packet is sent to the data transfer module of VCP where the IP-to-RMAC address binding occurs and a VCP/LLC/BMAC frame is transmitted to the base station network or to the backbone network, depending on whether the destination MH resides in this native virtual cell or moved to another virtual cell, respectively. If the IP packet is in transit, it is transmitted to the next-hop router.

The VCP/LLC/BMAC frame from the base station network is always sent to the VCP module. If the frame carries a VCARP packet, the address resolution module performs the IP-to-RMAC address binding with GMT and sends it back to the base station network. If the frame carries an IP packet with a BS's RMAC address of the virtual cell in the VCP header, the data transfer module dose not search GMT and directly pass the IP packet to the routing module where the next-hop router is determined. Otherwise, the data transfer module searches GMT with the destination

RMAC address and then transmits a VCP/LLC/BMAC frame back to the base station network or to an adjacent virtual cell following a forwarding pointer. At the same time, if the destination MH resides in this virtual cell, the ALS sends to the source BS a VCARP reply which conveys the destination BMAC address so that the subsequent data transfer can directly go to the destination BS.

3.3 Virtual Cell Protocol

3.3.1 Distributed Location Information

There are generally three different ways to distribute location information: centralized, partitioned, and duplicated. Depending on which way is used to treat location information, there is a tradeoff between location registration and paging. In a centralized system, a large volume of location updates at one physical site may degrade the network performance significantly. However, the consistency of location information is obtained and simple paging is achieved. A typical example is the first generation of mobile cellular telephone systems. In a partitioned system, the different partitions of location information are held at different sites. An example is IPIP where every BS maintains its local location information and paging is basically required when two remote physical cells are involved in communication unless a forwarding pointer is found. Thus, this approach can alleviate the problem of location registration in the centralized system, but may need frequent paging. In a duplicated system, on the other hand, the same location information is held at the different sites. VIP is an example where the location information of an MH is held on several intermediate gateways. Although it can give an optimal routing path in the normal case, when a communicating MH migrates, the inconsistent location information may be spread over several intermediate gateways.

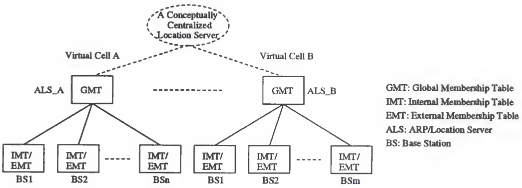


Figure 3.6. Distributed Location Information in the Virtual Cell System

In order to support mobility management in a distributed manner, the virtual cell takes advantage of the distributed hierarchical location information which involves a combination of partitioning, duplication, and centralization, as shown in Figure 3.6. GMT is partitioned between virtual cells. On the other hand, IMT is partitioned with other IMTs and duplicated with GMT in a virtual cell. EMT partially duplicates a part of GMT for MHs in different physical cells in the same virtual cell so that address resolution and location tracking for data transfer are first tried to accomplish among BSs independent of the ALS. Thus, GMT is only referred in case that they cannot be solved among BSs.

It should be noted that there is another conceptual hierarchy above the GMT level. When MH_1 migrates from its native virtual cell A to virtual cell B, forwarding pointers are maintained from ALS_A to ALS_B to the current BS at the VCP layer. From the prospective of the IP layer, however, MH_1 is regarded as if it were in its native virtual cell. Thus, when a remote fixed host or an MH in a third network wants to send a packet to MH_1, the existing routing protocols of the IP layer correctly deliver the packet to ALS_A. Then ALS_A traces MH_1 using the forwarding pointers. Therefore, the IP network address of a migrating MH represents at least the near-location information of the MH, and when coupled with the existing routing protocols

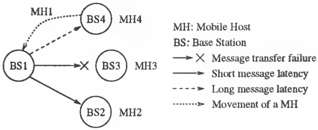


Figure 3.7. An Example of the Inconsistency Problem

of the IP layer it can give the same effect that we maintain a conceptually centralized server for the whole mobile network. If MH_1 migrates within a virtual cell, the IP network address of MH_1 gives the exact location information.

3.3.2 Impact of Mobility

The location registration due to the handoff procedure is a main source of inconsistent location information. To achieve the consistency of distributed information is not trivial at all and many researchers have been working on this problem in fixed network environments [17, 18, 19]. In the virtual cell environment, however, host mobility is integrated with fixed networks, which makes the problem even more complex. To solve this problem, the handoff procedure should utilize the multicasting ability of the base station network.

Assuming that the distributed location information is initially consistent, consider that MH_1 moves from BS_4 to BS_1 which periodically broadcasts its identity, as shown in the Figure 3.7. When MH_1 receives a beacon packet, it sends a greeting message containing its identity to BS_1. BS_1 detects the entry of MH_1 to its local cell, deletes its corresponding EMT entry if exists, and inserts a new IMT entry. Then, BS_1 multicasts a location registration message to every BS so that it maintains the consistent location information for MH_1. However, some BSs may fail to receive the message due to communication errors and buffer overflows. The long propagation

delay also has the similar effect as the message loss at a point in time due to mobility. To explore a practical solution, we should begin with the assumption that the base station network supports the atomic multicasting by using an existing protocol such as the Trans protocol [19]. The basic idea of the Trans protocol is that acknowledgements for multicast messages are piggybacked on messages that are themselves multicast, using a combination of positive and negative acknowledgement strategies. This piggyback feature is suitable for the handoff procedure because in the steady state, if some MHs move out then there will be a high possibility that other MHs move in.

Note that although the atomic multicast is supported, the message loss can still occur if BS_1 multicasts the location registration message. Assume that BS_3 failed to receive it and BS_4 received it with a long message latency. Then, the following four cases can happen when other MHs transmit a message to MH_1 during the process of the atomic multicast:

1. If BS_4 has the old location information for MH_1, the message transmitted from MH_4 to MH_1 will be lost.

2. If BS_4 has the new location information for MH_1, the message transmitted from MH_4 to MH_1 will be received.

3. Because BS_3 has the old location information for MH_1, the message transmitted from MH_3 to MH_1 will be lost or received, depending on what value BS_4 has. If it has the old value, the message will be lost. Otherwise, the message will be redirected to BS_1 by BS_4.

4. Because BS_2 has the new location information for MH_1, the message transmitted from MH_2 to MH_1 will be received.

From the above observation, we can see that at least the previous base station BS_4 must have the new location information for MH_1 as soon as possible in order to avoid a possible message loss or to mitigate the effect of long message latency.

3.3.3 Handoff Procedure

The new BS informs the previous BS of an MH's migration using point-to-point communication immediately after detecting the MH's identity so that all messages received at the previous BS during the handoff procedure can be correctly redirected to the new BS. Next, the previous BS is responsible for maintaining the consistency of the MH's location information at all BSs using an atomic multicasting. Note that the ALS is excluded from the multicasting group. If the ALS is involved, every movement of the MH will need an access to the ALS and there is no difference from the centralized system. However, it may cause another inconsistency problem between the ALS and BSs because the ALS may have old location information for some MHs. Thus, the previous BS periodically backups the location information updated during a predefined time interval to the ALS using point-to-point communication. When an MH continues to move to different physical cells, there can be a redirection chain with multiple hops from the first BS to the current BS. However, the chain is eliminated when the newest location information is received through an atomic multicasting.

Figure 3.8 shows a communication model for the handoff procedure. For convenient description, we define the following notations where the VCP header information is omitted:

- $\underline{A \Longrightarrow B, \ldots, N : \{\text{frame}\}}$ means that A multicasts {frame} to $B, \ldots, N$ over the base station network, where {frame} consists of {source address, destination address | frame data}.

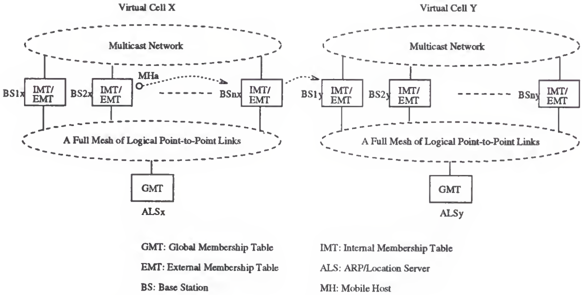


Figure 3.8. A Communication Model for Handoff Procedure

- $\underline{A \longrightarrow B : \{\text{frame}\}}$ means that A sends {frame} to B via a point-to-point link.
- $\underline{A \rightsquigarrow B : \{\text{frame}\}}$ means that A broadcasts {frame} whose destination is B via radio links.
- $\underline{IP(A)}$ denotes the IP address of an MH A.
- $\underline{RMAC(A)}$ denotes the radio MAC address of A, where A may be a BS or an MH.
- $\underline{BMAC(A)}$ denotes the address of A in the base station network (BMAC), where A may be a BS or an ALS.

Within a virtual cell. Consider that MH_a moves from BS_{2x} to BS_{nx}. In addition to its own addresses, $IP(MH_a)$ and $RMAC(MH_a)$, MH_a initially keeps the current base station addresses, $RMAC(BS_{2x})$ and $BMAC(BS_{2x})$.

1. $BS_{nx} \rightsquigarrow MH_a : \{RMAC(BS_{nx}),$ broadcast address $\mid BMAC(BS_{nx})\}$

MH_a receives a beacon including $BMAC(BS_{nx})$ from BS_{nx}, decides to switch from BS_{2x} to BS_{nx}, and updates its base station address from $RMAC(BS_{2x})$ and $BMAC(BS_{2x})$ to $RMAC(BS_{nx})$ and $BMAC(BS_{nx})$, respectively.

2. $MH_a \rightsquigarrow BS_{nx} : \{RMAC(MH_a),\ RMAC(BS_{nx}) \mid IP(MH_a),\ BMAC(BS_{2x})\}$

   BS_{nx} receiving a greeting message from MH_a processes its IMT and EMT using the source address $RMAC(MH_a)$ as a key. If the EMT entry for MH_a is found, the entry is deleted and then the IMT entry for MH_a is added. $IP(MH_a)$ in the EMT entry is used for fast address resolution, as described in Section 3.3.4.

3. $BS_{nx} \longrightarrow BS_{2x} : \{BMAC(BS_{nx}),\ BMAC(BS_{2x}) \mid$
   $IP(MH_a),\ RMAC(MH_a),\ BMAC(BS_{nx})\}$

   BS_{2x} receiving the notification of MH_a's migration sends an acknowledgement to BS_{nx} and then processes IMT and EMT. The IMT entry for MH_a is deleted and the new EMT entry for MH_a is added.

4. $BS_{2x} \Longrightarrow BS_{1x}, \ldots, BS_{nx} : \{BMAC(BS_{2x}),$ multicast address $\mid$
   $IP(MH_a),\ RMAC(MH_a),\ BMAC(BS_{nx})\}$

   Every BS receiving the notification of MH_a's migration except BS_{nx} processes its EMT. If the EMT entry for MH_a already exist, the entry is updated.

Between virtual cells. Consider that MH_a moves from BS_{nx} to BS_{1y}. In addition to its own addresses, $IP(MH_a)$ and $RMAC(MH_a)$, MH_a keeps the current base station addresses, $RMAC(BS_{nx})$ and $BMAC(BS_{nx})$.

1. $BS_{1y} \rightsquigarrow MH_a : \{RMAC(BS_{1y}),$ broadcast address $\mid BMAC(BS_{1y})\}$

2. $MH_a \rightsquigarrow BS_{1y} : \{RMAC(MH_a),\ RMAC(BS_{1y}) \mid IP(MH_a),\ BMAC(BS_{nx})\}$

BS_{1y} receiving a greeting message from MH_a detects from the previous base station address $BMAC(BS_{nx})$ that MH_a was in another virtual cell. BS_{1y} creates the IMT entry for MH_a.

3. $BS_{1y} \longrightarrow ALS_Y : \{BMAC(BS_{1y}),\ BMAC(ALS_Y) \mid$
   $IP(MH_a),\ RMAC(MH_a),\ BMAC(BS_{1y}), BMAC(BS_{nx})\}$

   ALS_Y receiving a location registration message creates the GMT entry for MH_a using $IP(MH_a)$, $RMAC(MH_a)$, and $BMAC(BS_{1y})$. Then, ALS_Y knows that BS_{nx} is a base station in virtual cell X, using $BMAC(BS_{nx})$.

4. $ALS_Y \longrightarrow ALS_X : \{BMAC(ALS_Y),\ BMAC(ALS_X) \mid$
   $IP(MH_a),\ RMAC(MH_a),\ BMAC(ALS_Y), B_MAC(BS_{nx})\}$

   ALS_X receiving a location registration message updates the GMT entry for MH_a using $IP(MH_a)$, $RMAC(MH_a)$, and $BMAC(ALS_Y)$. This entry serves as a forwarding pointer at the VCP layer if ALS_X is the ALS of MH_a's native virtual cell. If it is not, ALS_X informs the ALS of MH_a's native virtual cell of MH_a's migration to virtual cell Y so that a forwarding pointer is maintained from MH_a's native virtual cell to virtual cell Y.

5. $ALS_X \longrightarrow BS_{nx} : \{BMAC(ALS_X),\ BMAC(BS_{nx}) \mid$
   $IP(MH_a),\ RMAC(MH_a),\ BMAC(ALS_X)\}$

   BS_{nx} receiving the notification of MH_a's migration sends an acknowledgement to BS_{1y} in the reverse direction and then processes IMT and EMT. The existing IMT entry is deleted and an EMT entry using $IP(MH_a)$, $RMAC(MH_a)$, and $BMAC(ALS_X)$ is added.

6. $BS_{nx} \Longrightarrow BS_{1x}, BS_{2x}, \ldots, BS_{(n-1)x} : \{BMAC(BS_{nx}),$ multicast address $\mid$
   $IP(MH_a),\ RMAC(MH_a),\ BMAC(ALS_X)\}$

Every BS receiving the notification of MH_a's migration except BS_{nx} manipulates its EMT. If the EMT entry exists, it is updated.

3.3.4 Address Resolution

VCARP is designed to get the physical address of the MH in a remote physical cell of the virtual cell and should satisfy at least the following three requirements. First, in order to achieve the compatibility with ARP, VCARP must be shielded from MHs as if they were in a physical cell. Second, because the VCARP packet latency can directly affect the scalability of a virtual cell, a distributed address resolution mechanism should be considered rather than broadcasting or a centralized solution. Third, when an MH moves to an adjacent physical cell immediately after sending an ARP request and further moves to another physical cell, the ARP reply may be lost and then the ARP request may be repeatedly generated. Hence, VCARP must also support host mobility.

Figure 3.9 shows the ARP/VCARP packet format. The ARP packet follows exactly the same format as the existing ARP standard and has longer fields *SENDER HA* and *TARGET HA* in order to make it possible to accommodate the use of hierarchical 64-bit E.164 network addresses in radio networks. The VCARP packet format has an additional 64-bit field *BASE HA* which is used to convey the location information which is expected to be used for data transfer in the very near future. The source MH performs ARPing only when it resides in its native virtual cell and the destination MH belongs to the same native virtual cell. Otherwise, it directly transmits IP packets without ARPing.

Within a physical cell. The procedure is the same as in the normal ARP. The same ARP request/reply packet formats are used where an 8-bit hardware address

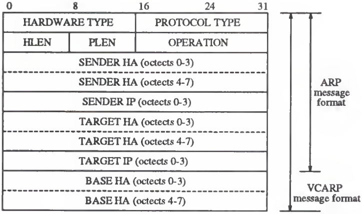
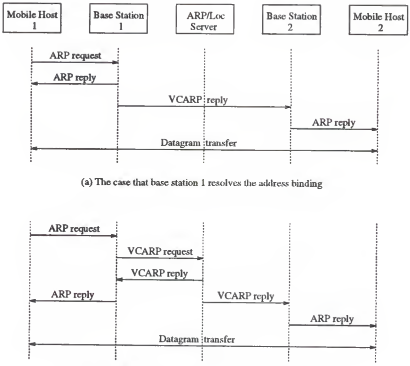


Figure 3.9. ARP/VCARP Packet Format

length field allows to accommodate arbitrary radio network addresses. The IP protocol of the source MH checks the destination IP address with its address resolution table. If the address binding is successful, the source MH uses the RMAC address to transfers datagrams. Otherwise, the source MH broadcasts an ARP request. As soon as the destination MH receives the ARP request, it responds with an ARP reply that includes the requested physical address of the destination. At the same time, the BS that received the ARP request searches IMT and knows that the destination MH is in the local physical cell so it discards the ARP request.

Within a virtual cell. Figure 3.10(a) shows the flow of messages when MH_1 broadcasts an ARP request to get the physical address of MH_2 and it is resolved at BS_1. BS_1 that received the ARP request searches its EMT. The EMT entry for MH_2 already contains its physical address and BS_1 directly sends an ARP reply within the physical cell. At the same time, BS_1 sends a VCARP reply to BS_2. Then, BS_2 transforms the format of the VCARP reply to that of the ARP reply by stripping out the field BASE HA and broadcasts the transformed ARP reply in its local cell. This VCARP reply also conveys the location information of MH_1 to BS_2 through the field BASE HA which contains the network address of BS_1. Then, BS_2



(a) The case that base station 1 resolves the address binding

(b) The case that the ARP/Location server resolves the address binding

Figure 3.10. Virtual Cell Address Resolution Protocol (VCARP) Flow Diagram

creates an EMT entry so that MH_2 can directly transmit data to MH_1 without any additional address resolution in the near future communication. It is based on the observation that computer communication is usually bidirectional; if MH_1 has some reason to talk to MH_2, then MH_2 will probably have some reason to talk to MH_1.

Figure 3.10(b) shows the flow of messages when the address resolution is accomplished at the ALS. BS_1 sends a VCARP request to the ALS because it has no entry for MH_2. Then, the ALS sends a VCARP reply to BS_1 after resolving MH_2's address binding using GMT. At the same time, the ALS also sends BS_2 a VCARP reply with the same reason as in the above case. The location information of MH_1 and MH_2 is also conveyed via both VCARP replys to establish a logical bidirectional link between them. Now we describe the support of host mobility in VCARP. Consider the case that MH_1 requested an address resolution to send datagrams to MH_2 and then moved to a third BS_n before the VCARP reply is arrived. By the handoff procedure, BS_1 knows that MH_1 currently belongs to BS_n. As soon as BS_1 receives the VCARP reply, it redirects the packet to BS_n.

3.3.5 Data Transfer

The transmission of packets between virtual cells and fixed networks have been described in the previous sections. This section deals with the packet transmission within a physical cell and within a virtual cell. Let BS_{ij} denote the i-th BS of virtual cell j in Figure 3.6, where $i = 1, \ldots, n$ when $j = A$ and $i = 1, \ldots, m$ when $j = B$. We assume that MH_{ij} belongs to BS_{ij} and M_x initially belongs to BS_{1A}. We also assume that the address binding is achieved by VCARP.

Within a physical cell. Consider the case that MH_{1A} wants to send a packet to MH_x. By the address resolution protocol, MH_{1A} knows the RMAC address of MH_x and directly broadcasts the packet to it. BS_{1A} knows that MH_x resides in its physical

cell by checking IMT using the RMAC address of MH_x as a key, and discards the received packet.

Within a virtual cell. Consider the case that MH_x moves from BS_{1A} to BS_{2A} and each of MH_{1A}, MH_{2A}, and MH_{nA} wants to send a packet to MH_x. By the handoff procedure, a forwarding pointer is set from BS_{1A} to BS_{2A} at the EMT entry of BS_{1A} and the packet from MH_{1A} is correctly delivered to MH_x by the forwarding pointer. The packet from MH_{2A} is directly delivered to MH_x and BS_{2A} discards it because the IMT entry for MH_x is found. The delivery of the packet from MH_{nA} has three cases. If the EMT entry for MH_x is found before the completion of BS_{1A}'s multicasting operation, the packet will be delivered to BS_{2A} via BS_{1A}. If the EMT entry is found after the completion of its multicasting operation, the packet will be directly delivered to BS_{2A}. If the EMT entry is not found, the first packet from MH_{nA} will be delivered to BS_{2A} through the ALS. At the same time, the ALS generates a VCARP reply in order to have BS_{nA} learn that MH_x is currently belongs to BS_{2A}. It makes it possible to directly transfer the subsequent packets following the first one to BS_{2A} without going through the ALS repeatedly.

3.4 Comparison and Discussion

Figure 3.11 summarizes the characteristics of the three representative mobile data networks and compares them with those of the virtual cell system. When considering the ubiquitous coverage of TCP/IP networks and applications, the virtual cell has several advantages against IP-level mobile host protocols. Because the virtual cell uses a underlying datagram network to interconnect BSs and to form flexible coverage, the difficulties arising from diverse underlying networks can be avoided in terms of the performance of mobility management function and the complexity of network management function. Moreover, because the virtual cell is a logical cell and the

Networks / items	IPIP	VIP	CPS	VCP
Target Applications	TCP/IP applications	TCP/IP applications	packetized voice and data services	TCP/IP applications
Target environment	campus area	wide area	metropolitan area	wide area
Base station networks	point-to-point physical links (Internet)	point-to-point physical links (Internet)	DQDB MAN	packet-switched connectionless net with multicast ability (SMDS)
Mobility support layer	network layer	network layer	data link layer	data link layer
Type of movement	mobility	portablility/ mobility	mobility	mobility
Type of packet switching	datagram	datagram	virtual circuit	datagram
Address resolution	Proxy ARP	address mapping	no	VCARP
Scalability	low	high	high	high
Cell coverage area	physically bounded	physically bounded	physically bounded	logically flexible

Figure 3.11. Comparison of the Virtual Cell System With Other Networks

bridge function is usually much faster than the gateway function, it is possible to increase performance significantly if we can properly engineer the deployment of virtual cells so that the intra virtual cell traffic is as much larger than the inter virtual cell traffic as possible. Preserving the interconnection level of BSs at the data link layer also makes it possible to shield host mobility from the IP layer, and using the base station network address as a contact point of an MH makes it possible to avoid acquiring a temporary address by some kind of local utilities whenever the MH migrates. Thus, the problems of interfering with the conventional IP layer and of reserving a large amount of IP address space for location information can be solved.

There are two major issues related to the performance of VCP. Within a virtual cell, a BS first tries to perform address resolution and data transfer locally without the intervention of the ALS, based on its two membership tables, IMT and EMT. Thus, the hit ratio of the EMT search directly affect the degree of referring to the

ALS when remote MHs in different physical cells are involved in communication. This issue directly addresses the size and update scheme of EMT, and it is closely related to the IP traffic and host mobility pattern. A number of research results on the internet protocol traffic analysis in fixed network environment reveals that certain hosts communicate more with one another than with other hosts [39, 21, 22]. Even in the virtual cell environment, the locality of destination hosts can have an important implication for each of the address resolution and data transfer functions. A small ARP cache at each MH can greatly reduce VCARP traffic within a virtual cell. The address resolution should not severely burden the size of EMT. If we do not consider host mobility, the locality property also means that data transfer does not severely burden the size of EMT. When considering host mobility, the update scheme of EMT could be rather important if we can properly deploy virtual cells based on mobility and traffic patterns among physical cells.

The other issue is the amount of multicast traffic among BSs, which is generated only by the handoff function of the virtual cell. Because the multicasting ability is supported by switches in the base station network, each BS is not related to the load of generating the multicast traffic, and the bandwidth usage of the base station network is affected by it. The relatively short length of the handoff message should not severely affect the high-bandwidth base station network. This will be addressed in Chapter 6.

3.5 Chapter Summary

We have presented a new network infrastructure for the transmission of IP datagrams in mobile computer communications. With the virtual cell concept, physical cells are grouped into larger logical cells where host mobility is shielded from the IP layer. It eliminates the need of IP-level protocols for host mobility within a virtual

cell and provides the flexible coverage area that can be properly engineered according to mobility and communication patterns among physical cells. In order to achieve this concept, we have described the virtual cell protocol that consists of handoff, address resolution, and data transfer modules, based on the distributed hierarchical location information of mobile hosts.

Given mobility and communication patterns among physical cells, the next chapter deals with the problem of deploying virtual cells in highway cellular systems. The problem is transformed to an optimization problem of finding a cover of disjoint clusters of physical cells, where each cluster corresponds to a virtual cell.

CHAPTER 4
OPTIMAL PARTITIONING IN HIGHWAY CELLULAR SYSTEMS

Given a linear array of n base stations which generate multiple types of traffic among themselves, we consider the problem of finding a cover of disjoint clusters of neighboring base stations. The objective is to minimize the total communication cost for the entire system where the cost of intra-cluster communication is lower than that of inter-cluster communication for each type of traffic. The optimization problem is transformed into the dual based on the relative cost that is the communication cost if a pair of nodes are in different clusters of a partition and reflects multiple types of traffic with multiple dynamic cost functions. Using the relative cost matrix, an efficient algorithm of $O(mn^2)$, where m is the number of clusters in a partition, is designed by dynamic programming.

4.1 Problem Formalization

4.1.1 Problem Statement

The base stations in a highway cellular system are ordered linearly as stations $1, 2, \ldots, n$. We are interested in finding clusters of neighboring stations, such that a location server can be allocated to each cluster. In other words, the partition of base stations must follow the underlying topology constraint, i.e., a linear arrangement in highway cellular systems.

The communication among base stations is considered as a full mesh of point-to-point logical network. This network represents a possible communication between mobile hosts through different base stations. The network is described by a complete

directed graph $G = (V, E)$, where $|V| = n$. The vertices of the graph represent the base stations of the network, and their indices are numbered from 1 to n in sequence for n concatenated cells. The edges represent directional communication links between vertices, and each of which is assigned a move frequency by a function $f_m : V \times V \to R^+$ and a find frequency by $f_f : V \times V \to R^+$. Denote w_m^1 and w_m^2 as the weight of a move operation within a cluster and between clusters, respectively, and w_f^1 and w_f^2 as that of a find operation within a cluster and between clusters, respectively. Then we define a cost function $c_i^j : V \times V \to R^+$, where i = m or f and $j = 1$ or 2, such that for all $(u, v) \in V \times V$,

$$c_i^j(u,v) \quad = \quad f_i(u,v)w_i^j. \tag{4.1}$$

The set of equations given by (4.1) implies that each edge is dynamically assigned one of two different costs for each type of operation, depending on whether it is involved in a cluster or between clusters. Given the number of clusters m for $1 \le m \le n$, a sequence $\{1, \ldots, n\}$ of n vertices can be linearly partitioned into $\Pi = \{P_1, \ldots, P_m\}$ such that $P_i \cap P_j = \emptyset$ and $\cup_i P_i = V$, where $i, j = 1, \ldots, m$ and $i \neq j$. The intra-cluster communication cost of Π is then

$$Cost_{intra}(\Pi) = \sum_{i=1}^{m} \sum_{u,v \in P_i} (c_m^1(u,v) + c_f^1(u,v)) \tag{4.2}$$

and the inter-cluster communication cost of Π is

$$Cost_{inter}(\Pi) = \sum_{i=1}^{m-1} \sum_{u \in P_i, v \in P_j, i<j} ((c_m^2(u,v) + c_f^2(u,v)) + (c_m^2(v,u) + c_f^2(v,u))). \tag{4.3}$$

From (4.2) and (4.3), the total communication cost of Π is therefore

$$Cost_{total}(\Pi) = Cost_{intra}(\Pi) + Cost_{inter}(\Pi). \tag{4.4}$$

Figure 4.1 illustrates a network model of size 6 with two frequency matrices, f_m and f_f. When $w_m^1 = w_f^1 = 1$, $w_m^2 = 3$, and $w_f^2 = 2$, the cost matrices of Figure 4.2 are

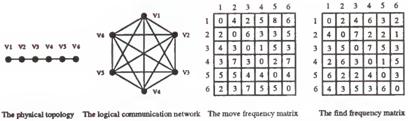


Figure 4.1. An Example of the Network Model

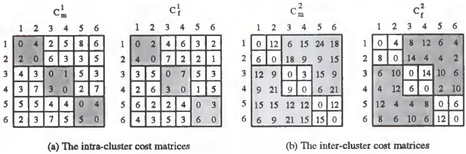


Figure 4.2. Intra- and Inter-cluster Cost Matrices

driven from (4.1). If a partition $\Pi = \{P_1, P_2, P_3\}$, where $P_1 = \{1,2\}$, $P_2 = \{3,4\}$ and $P_3 = \{5,6\}$, $Cost_{intra}(\Pi)$ and $Cost_{inter}(\Pi)$ is the sum of all elements in the shaded areas of Figure 4.2 (a) and (b), respectively. Thus, $Cost_{total}(\Pi) = Cost_{intra}(\Pi) + Cost_{inter}(\Pi) = 44 + 493 = 537$.

Given n and m, our task is to find an optimal partition Π which minimizes $Cost_{total}$ subject to $1 \leq |P_i| \leq b \leq n$ for every $i = 1, \ldots, m$, where $|P_i|$ is the number of vertices in the ith cluster. Since the cluster size constraint does not add the time and space complexity and affect our algorithm, we first present our solution for the general case without the constraint, and then we show that the solution is simply modified to consider the constraint. Before presenting our solution, let us point out that exhaustively checking all possible partitions does not yield an efficient algorithm. For example, if an ordered sequence $\{1, 2, 3, 4\}$ of 4 vertices is given, we can partition it in 8 distinct ways:

$\{\{1, 2, 3, 4\}\}$ if $m = 1$;

Table 4.1. Number of Partitions

n	The number of clusters m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8	m=9
1	1								
2	1	1							
3	1	2	1						
4	1	3	3	1					
5	1	4	6	4	1				
6	1	5	10	10	5	1			
7	1	6	15	20	15	6	1		
8	1	7	21	35	35	21	7	1	
9	1	8	28	56	70	56	28	8	1

{{1, 2, 3}, {4}}, {{1, 2}, {3, 4}}, and {{1}, {2, 3, 4}} if m = 2;

{{1, 2}, {3}, {4}}, {{1}, {2, 3}, {4}}, and {{1}, {2}, {3, 4}} if m = 3;

and {{1}, {2}, {3}, {4}} if m = 4.

Table 4.1 gives the number of ways to partition a sequence $\{1, 2, \dots, n\}$ of n vertices into m clusters for $1 \leq m \leq n$, preserving order. Since the numbers in Table 4.1 form Pascal's triangle, if given n and m, the number of partitions is $G(n,m) = \binom{n-1}{m-1}$. For example, if $n = 9$ and $m = 5$, $G(9,5) = \binom{8}{4} = 70$.

Because row n of Table 1 sums to 2^{n-1} by the binomial theorem, the total number of partitions of n ordered vertices is 2^{n-1} for $n \geq 1$. Thus, the time complexity of exhaustive search is $O(2^n n^2)$ and its space complexity is $O(n^2)$ for every m. Given n and m, it has $O(G(n,m)n^2)$ time complexity and $O(n^2)$ space complexity. The brute-force method of exhaustive search is therefore a poor strategy for determining an optimal partition. The results of our algorithm by dynamic programming based on a relative cost matrix show $O(n^3)$ time complexity and $O(n^2)$ space complexity for every m, and $O(mn^2)$ time complexity and $O(n)$ space complexity for each m.

4.1.2 Dual Problem

The dynamic property of multiple cost functions for multiple types of traffic not only makes the optimization problem complex in terms of computation and storage but also makes it difficult to design an efficient algorithm. It is therefore desirable to express the optimization problem with multiple cost functions as an equivalent problem with one cost function without affecting the objective. Thus, we combine the set of equations given by (4.1) into a relative cost function $g : V \times V \rightarrow R^+$ such that for all $(u, v) \in V \times V$,

$$g(u,v) = f_m(u,v)(w_m^2 - w_m^1) + f_f(u,v)(w_f^2 - w_f^1), \tag{4.5}$$

where $(w_m^2 - w_m^1)$ is the relative weight of a move operation and $(w_f^2 - w_f^1)$ is the relative weight of a find operation. $g(u,v)$ then represents the total relative cost of $f_m(u,v)$ move and $f_f(u,v)$ find operations from vertex u to vertex v if they are in different clusters. Using the equation (4.5), the dual to the optimization problem can now be written as

$$\begin{array}{ll} \text{Min.} & \sum_{i=1}^{m-1} \sum_{u\in P_i, v\in P_j, i<j} (g(u,v) + g(v,u)) \\ \text{s.t.} & 1 \leq |P_i| \leq b \leq n, \quad i = 1, \ldots, m, \end{array}$$

where $|P_i|$ is the number of vertices in the ith cluster. Let us now define

$$Rcost_{inter}(\Pi) = \sum_{i=1}^{m-1} \sum_{u\in P_i, v\in P_j, i<j} (g(u,v) + g(v,u)), \tag{4.6}$$

$$Upper_{intra}(\Pi) = \sum_{i=1}^{m-1} \sum_{u\in P_i, v\in P_j, i<j} (c_m^1(u,v) + c_f^1(u,v)), \tag{4.7}$$

$$Lower_{intra}(\Pi) = \sum_{i=1}^{m-1} \sum_{u\in P_i, v\in P_j, i<j} (c_m^1(v,u) + c_f^1(v,u)). \tag{4.8}$$

The correctness of the transformation is then proved by the lemma and theorem below.

Lemma 4.1 Let $Cost_{constant}(\Pi) = Cost_{intra}(\Pi) + Upper_{intra}(\Pi) + Lower_{intra}(\Pi)$. Then $Cost_{total}(\Pi) = Cost_{constant}(\Pi) + Rcost_{inter}(\Pi)$.

Proof of Lemma 4.1 From (4.1), (4.2), (4.3) and (4.4),

$$\begin{aligned}
Cost_{total}(\Pi) &= \sum_{i=1}^{m} \sum_{u,v \in P_i} (f_m(u,v)w_m^1 + f_f(u,v)w_f^1) + \\
&\quad \sum_{i=1}^{m-1} \sum_{u \in P_i, v \in P_j, i<j} (f_m(u,v)w_m^2 + f_f(u,v)w_f^2) + \\
&\quad \sum_{i=1}^{m-1} \sum_{u \in P_i, v \in P_j, i<j} (f_m(v,u)w_m^2 + f_f(v,u)w_f^2). \qquad (4.9)
\end{aligned}$$

Adding and subtracting the same value to and from the right-hand side of (4.9) still hold the equality. Thus, by adding $Upper_{intra}(\Pi)$ (4.7) and $Lower_{intra}(\Pi)$ (4.8) to the first term of the right-hand side of (4.9), and by subtracting $Upper_{intra}(\Pi)$ from the second term and $Lower_{intra}(\Pi)$ from the third term, we obtain

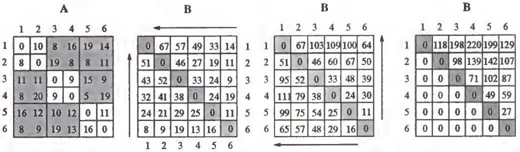
$$\begin{aligned}
Cost_{total}(\Pi) &= Cost_{constant}(\Pi) + \\
&\quad \sum_{i=1}^{m-1} \sum_{u \in P_i, v \in P_j, i<j} (f_m(u,v)(w_m^2 - w_m^1) + f_f(u,v)(w_f^2 - w_f^1)) + \\
&\quad \sum_{i=1}^{m-1} \sum_{u \in P_i, v \in P_j, i<j} (f_m(v,u)(w_m^2 - w_m^1) + f_f(v,u)(w_f^2 - w_f^1)). (4.10)
\end{aligned}$$

Hence, applying (4.5) and (4.6) to (4.10) in sequence,

$$\begin{aligned}
Cost_{total}(\Pi) &= Cost_{constant}(\Pi) + \sum_{i=1}^{m-1} \sum_{u \in P_i, v \in P_j, i<j} (g(u,v) + g(v,u)) \\
&= Cost_{constant}(\Pi) + Rcost_{inter}(\Pi).
\end{aligned}$$

□

Figure 4.3(a) shows the relative cost matrix driven from the move and find frequency matrices in Figure 4.1 using (4.5) if $w_m^2 - w_m^1 = 2$ and $w_f^2 - w_f^1 = 1$. For the partition $\Pi = \{\{1,2\},\{3,4\},\{5,6\}\}$, therefore, the dual problem is concerned with the sum of all elements in the shaded areas, say, $Rcost_{inter}(\Pi) = 300$. Since



(a) relative cost matrix (b) transformation process (c) cumulative cost matrix

Figure 4.3. Constructing the Cumulative Cost Matrix

$Cost_{constant}(\Pi) = Cost_{intra}(\Pi) + Upper_{intra}(\Pi) + Lower_{intra}(\Pi) = 44 + 96 + 97 = 237$, $Cost_{total}(\Pi) = Cost_{constant}(\Pi) + Rcost_{inter}(\Pi) = 237 + 300 = 537$.

Theorem 4.1 If Π is an optimal partition which minimizes $Rcost_{inter}$, then Π is also an optimal partition which minimizes $Cost_{total}$.

Proof of Theorem 4.1 By Lemma 4.1, it follows that $Cost_{total}(\Pi) = Cost_{constant}(\Pi) + Rcost_{inter}(\Pi)$. Note that $Cost_{constant}(\Pi)$ is constant independent of how to partition because it consists of three intra-cluster communication costs defined in the equations (4.2), (4.7), and (4.8). Thus, it holds that Π minimizes $Rcost_{inter}$ iff Π minimizes $Cost_{total}$. $\square$

4.1.3 Cumulative Cost Matrix

We have the relative cost matrix $\mathbf{A}$ of size n such that $\mathbf{A}(u,v) = g(u,v)$ for all $(u,v) \in V \times V$, as shown in Figure 4.3(a). The dual problem can then be restated as the problem of partitioning $\mathbf{A}$ into submatrices $\mathbf{A}_{ij}$ for $i,j = 1,\ldots,m$ such that the main diagonal submatrices $\mathbf{A}_{ii}$ are square and

$$Rcost^{A}_{inter}(\Pi) = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} (\mathbf{A}^*_{ij} + \mathbf{A}^*_{ji}) \tag{4.11}$$

is minimized, where $\mathbf{A}^*_{ij}$ is the sum of all elements in $\mathbf{A}$ which represents the relative cost from the ith cluster to the jth cluster. In the example of Figure 4.3(a) where

$P_1 = \{1,2\}$, $P_2 = \{3,4\}$, and $P_3 = \{5,6\}$, $Rcost^A_{inter}(\Pi) = \mathbf{A}^*_{12} + \mathbf{A}^*_{13} + \mathbf{A}^*_{23} + \mathbf{A}^*_{21} + \mathbf{A}^*_{31} + \mathbf{A}^*_{32} = 51 + 52 + 48 + 50 + 45 + 54 = 300 = Rcost_{inter}(\Pi)$.

Because $\mathbf{A}^*_{ij}$ and $\mathbf{A}^*_{ji}$ vary for each partition, it is desirable to obtain their values directly without duplicated computations. Thus, we transform the relative cost matrix $\mathbf{A}$ to a cumulative cost matrix $\mathbf{B}$ so that for an arbitrary partition it is possible to obtain the value of the expression $\sum_{j=i+1}^{m}(\mathbf{A}^*_{ij} + \mathbf{A}^*_{ji})$ of (4.11) by simply accessing to an element in $\mathbf{B}$ which is defined as:

$$\mathbf{B}(s,t) = \sum_{k=s}^{t-1}\sum_{l=t}^{n} \mathbf{A}(k,l) \quad \text{if } s < t; \tag{4.12}$$

$$\mathbf{B}(t,s) = \sum_{k=s}^{t-1}\sum_{l=t}^{n} \mathbf{A}(l,k) \quad \text{if } s < t. \tag{4.13}$$

The $\mathbf{B}$ matrix can be computed by accumulating the elements above the main diagonal of $\mathbf{A}$ from right to left for each row and then from bottom to top for each column, while the elements below the main diagonal of $\mathbf{A}$ are accumulated from bottom to top for each column and then from right to left for each row, as shown in Figure 4.3(b).

Lemma 4.2 Let the size of P_i be d_i such that $\sum_{i=1}^{m} d_i = n$. Given an integer i for $1 \le i \le m-1$,

a) $\mathbf{B}(s,t) = \sum_{j=i+1}^{m} \mathbf{A}^*_{ij}$ if $s < t$, where $s = d_1 + \cdots + d_{i-1} + 1$ and $t = d_1 + \cdots + d_{j-1} + 1$.

b) $\mathbf{B}(t,s) = \sum_{j=i+1}^{m} \mathbf{A}^*_{ji}$ if $s < t$, where $s = d_1 + \cdots + d_{i-1} + 1$ and $t = d_1 + \cdots + d_{j-1} + 1$.

Proof of Lemma 4.2 The equation (4.12) can be divided into $(m - i + 1)$ terms

$$\begin{aligned} \mathbf{B}(s,t) &= \sum_{k=s}^{t-1}\sum_{l=t}^{n} \mathbf{A}(k,l) \\ &= \sum_{k=s}^{t-1}\Big(\sum_{l=t}^{t+d_{j+1}-1} \mathbf{A}(k,l) + \sum_{l=t+d_{j+1}}^{t+d_{j+1}+d_{j+2}-1} \mathbf{A}(k,l) + \cdots + \sum_{l=t+d_{j+1}+\cdots+d_{m-1}}^{n} \mathbf{A}(k,l)\Big). \end{aligned}$$

Since each term is the sum of all elements in $\mathbf{A}_{ij}$, where $j = i+1, \ldots, m$, the left-hand side of a) can be expressed as

$$\begin{aligned} \mathbf{B}(s,t) &= \mathbf{A}^*_{i(i+1)} + \mathbf{A}^*_{i(i+2)} + \cdots + \mathbf{A}^*_{im} \\ &= \sum_{j=i+1}^{m} \mathbf{A}^*_{ij}. \end{aligned}$$

Proving in the same way as a), it is obvious that b) holds. □

Given the ith cluster P_i, Lemma 4.2 shows that for all $j = i+1, \ldots, m$, $\mathbf{B}(s,t)$ represents the sum of the relative cost from P_i to P_j, and $\mathbf{B}(t,s)$ represents the sum of the relative cost from P_j to P_i. Thus, $\mathbf{B}(s,t) + \mathbf{B}(t,s)$ gives the total relative cost among them. Since the positions of $\mathbf{B}(s,t)$ and $\mathbf{B}(t,s)$ are symmetric over the main diagonal, we combine them by adding $\mathbf{B}(t,s)$ to $\mathbf{B}(s,t)$ without affecting the optimization goal. Then, $\mathbf{B}(s,t) = b_{st} > 0$ if $s < t$ and $\mathbf{B}(s,t) = 0$ if $s \geq t$. Hence, the equation (4.11) can be rewritten by

$$Rcost^B_{inter}(\Pi) = \sum_{i=1}^{m-1} b_{st}, \tag{4.14}$$

where $s = \sum_{k=1}^{i} d_{k-1} + 1$, $d_0 = 0$ and $t = \sum_{k=1}^{i} d_k + 1$ for every i. In the example of Figure 4.3(c), $Rcost^B_{inter}(\Pi) = b_{13} + b_{35} = 198 + 102 = 300 = Rcost^A_{inter}(\Pi)$. The b_{13} entry is the cost between $\{1,2\}$ and $\{3,4,5,6\}$, and the b_{35} entry is that between $\{3,4\}$ and $\{5,6\}$. Thus, the sum of $b_{13} + b_{35}$ is the total cost among $\{1,2\}$, $\{3,4\}$, and $\{5,6\}$.

4.2 Algorithm

4.2.1 Optimal Structure and Overlapping Subproblems

Given the cumulative cost matrix $\mathbf{B} = (b_{ij})$, $1 \leq i < j \leq n$, we wish to find a sequence of $(m-1)$ elements whose sum is minimum such that the first element is chosen among b_{1j}, and if b_{ij} is selected as the rth element, the next possible element is one of

b_{jk}'s for $j+1 \leq k \leq n-(m-1-r)+1$. We denote the problem to yield this optimal sequence as LPART$(1, n, m)$. Let $< b_{1j}, b_{jk}, \ldots, b_{st} >$ be an optimal sequence of $(m-1)$ elements for the problem LPART$(1, n, m)$. Then, $< b_{jk}, \ldots, b_{st} >$ must be an optimal sequence of the subproblem LPART$(j, n, m-1)$. If it is not, there is another sequence $< b_{jl}, \ldots, b_{xy} >$ of $(m-2)$ elements whose sum is less than that of $< b_{jk}, \ldots, b_{st} >$. Hence, $< b_{1j}, b_{jl}, \ldots, b_{xy} >$ is a sequence of $(m-1)$ elements with less value, which is a contradiction. Thus, the LPART problem has an optimal-substructure property. In addition to the optimal-substructure property, the LPART problem has overlapping subproblems. LPART(j, n, p) for $1 < p < m$ is referenced $G(j-1, m-p)$ times during the computations of LPART$(j', n, p+1)$ for $j' = 1, 2, \ldots, j-1$ if $p \leq m-1$. For example, consider that $n = 9$ and $m = 5$. LPART(5,9,3) is referenced $G(4, 2) = \binom{3}{1} = 3$ times during the computations of LPART(4,9,4), LPART(3,9,4), and LPART(2,9,4) which correspond to the way of partitioning vertices 1 through 4 into $\{\{1, 2, 3\}, \{4\}\}$, $\{\{1, 2\}, \{3, 4\}\}$, and $\{\{1\}, \{2, 3, 4\}\}$, respectively. Thus, many other subproblems share subsubproblems. Since an optimal solution has optimal structure and overlapping subproblems, dynamic programming should be applied to the optimization problem.

4.2.2 Recursive Definition

We now define the value of an optimal solution recursively. Let $\phi_j(p)$ be the optimal value of the problem when a sequence $\{j, j+1, \ldots, n\}$ of vertices is linearly partitioned into p clusters. Then, we wish to find $\phi_1(m)$. The optimal cost $\phi_j(p)$ can be defined as follows. If $p = 1$, vertices j through n are grouped into one cluster, so no inter-cluster cost is necessary. If $p \geq 2$, each b_{jk} for $j+1 \leq k \leq n-p+2$ is considered as the first element, representing the cost between the first cluster $\{j, j+1, \ldots, k-1\}$ and the remaining $(p-1)$ clusters consisting of vertices k through n. Thus, the optimal

cost of partitioning $\{j, j+1, \ldots, n\}$ into p clusters is the minimum value among the sums of b_{jk} and $\phi_k(p-1)$ for all possible k. The dynamic programming formulation can be written as

$$\phi_j(p) = \begin{cases} 0, & \text{if} \quad p = 1, \\ \min_{j+1 \leq k \leq n-p+2} \{b_{jk} + \phi_k(p-1)\}, & \text{if} \quad p \geq 2, \end{cases} \tag{4.15}$$

for $1 \leq j \leq n-p+1$. From now on, we assume that each of j and k has the same range as in this section if it is not specified explicitly.

4.2.3 Computing Optimal Costs

Based on the recurrence relation (4.15), we present an algorithm to compute the optimal cost by using a bottom-up approach. The following pseudocode inputs the cumulative cost matrix $\mathbf{B} = (b_{ij})$ for $1 \leq i < j \leq n$ and the number of clusters m. During the execution of the procedure for each p, $1 \leq p \leq m$, the pseudocode uses the main diagonal element b_{jj} for storing the optimal cost $\phi_j(p)$ and the lower triangle element $b_{(n-p+2)j}$ for recording which index of k achieved the optimal cost $\phi_j(p)$.

```
LPART(1, n, m)
for j ← 1 to n
     b_jj ← 0
for p ← 2 to m
     for j ← n − p + 1 downto 1
          begin
               b_jj ← ∞
               for k ← j + 1 to n − p + 2
                    begin
                         sum ← b_jk + b_kk
                         if sum < b_jj
                              begin
                                   b_jj ← sum
                                   b_(n−p+2)j ← k
                              end
                    end
          end
```

	1	2	3	4	5	6
1	0	118	198	220	199	129
2	0	0	98	139	142	107
3	0	0	0	71	102	87
4	0	0	0	0	49	59
5	0	0	0	0	0	27
6	0	0	0	0	0	0

p = 1

	1	2	3	4	5	6
1	118	118	198	220	199	129
2	0	98	98	139	142	107
3	0	0	71	71	102	87
4	0	0	0	49	49	59
5	0	0	0	0	27	27
6	2	3	4	5	6	0

p = 2

	1	2	3	4	5	6
1	216	118	198	220	199	129
2	0	169	98	139	142	107
3	0	0	120	71	102	87
4	0	0	0	76	49	59
5	2	3/5	4	5	27	27
6	2	3	4	5	6	0

p = 3

	1	2	3	4	5	6
1	287	118	198	220	199	129
2	0	215	98	139	142	107
3	0	0	147	71	102	87
4	2	4	4	76	49	59
5	2	3/5	4	5	27	27
6	2	3	4	5	6	0

p = 4

	1	2	3	4	5	6
1	333	118	198	220	199	129
2	0	245	98	139	142	107
3	2	3	147	71	102	87
4	2	4	4	76	49	59
5	2	3/5	4	5	27	27
6	2	3	4	5	6	0

p = 5

	1	2	3	4	5	6
1	363	118	198	220	199	129
2	2	245	98	139	142	107
3	2	3	147	71	102	87
4	2	4	4	76	49	59
5	2	3/5	4	5	27	27
6	2	3	4	5	6	0

p = 6

Figure 4.4. Computation of the LPART(1,6,p) Problem for $1 \leq p \leq 6$

The algorithm first computes $b_{jj} \leftarrow 0$ for $j = 1, \ldots, n$ in lines 1-2, each represents the minimum cost of partitioning $\{j, \ldots, n\}$ into one cluster, $\phi_j(1)$. It then uses the recurrence relation (4.15) to compute the minimum cost of partitioning $\{j, \ldots, n\}$ into two clusters, $\phi_j(2)$, during the first execution of the loop in lines 3-16, and so forth. At each execution of the loop, the optimal cost is computed for $1 \leq j \leq n - p + 1$, assuming that the size of a cluster is at least 1. For a choice of p and j, the b_{jj} cost computed in lines 5-16 depends on the sum of the b_{jk} cost and the b_{kk} cost which was already computed from the previous execution of the loop in lines 3-16. Once k is determined, it is recorded at the $b_{(n-p+2)j}$ entry.

Figure 4.4 illustrates this procedure for $1 \leq p \leq 6$ when $n = 6$. The shaded entries in the table for a particular p are computed from the upper triangle and the main diagonal of the table for $(p-1)$. Note that the computation is practically achieved using only one table and each separate table is used to show the execution step as

p increases. The upper triangle containing the cumulative costs among vertices does not vary and the lower triangle is initially set to 0. When $p = 1$, every main diagonal element is set to 0. For the case of $p \neq 1$, consider how to compute the b_{11} entry when $p = 3$, which is equivalent to $\phi_1(3)$.

$$\phi_1(3) \;=\; \min\{b_{12} + \phi_2(2), b_{13} + \phi_3(2), b_{14} + \phi_4(2), b_{15} + \phi_5(2)\}$$

From the table for $p = 2$,

$$\begin{aligned}\phi_1(3) &= \min\{b_{12} + b_{22}, b_{13} + b_{33}, b_{14} + b_{44}, b_{15} + b_{55}\} \\ &= \min\{118 + 98, 198 + 71, 220 + 49, 199 + 27\} \\ &= 216\end{aligned}$$

when $k = 2$. Thus, $b_{11} = 216$ and $b_{51} = 2$.

4.2.4 Constructing an Optimal Partition

Since LPART$(1, n, m)$ only gives the optimal cost $\phi_1(m)$, we need to construct an optimal set of m clusters from the recorded k information in the lower triangle. Let $< k_1, k_2, \ldots, k_{m-1} >$ be a sequence of values of k for an optimal set of m clusters. The $b_{(n-m+2)1}$ entry gives the k_1 value, and the $b_{(n-(m-1)+2)k_1}$ entry gives the k_2 value, and the $b_{(n-(m-2)+2)k_2}$ entry gives the k_3 value, and so forth. Once $< k_1, k_2, \ldots, k_{m-1} >$ is obtained, we can easily divide the vertex set into m clusters, $\{1, \ldots, k_1 - 1\}$, $\{k_1, \ldots, k_2 - 1\}$, $\ldots$, $\{k_{m-1}, \ldots, n\}$, respectively. This is computed by the following pseudocode LPART-CLUSTER$(1, n, m)$.

LPART-CLUSTER$(1, n, m)$

LPART$(1, n, m)$
$CLUSTER \leftarrow \emptyset$
$k_0 \leftarrow 1$
$i \leftarrow n - m + 2$
$j \leftarrow 1$

```
6  for l ← 1 to m − 1
7      begin
8          k_l ← b_ij
9          CLUSTER ← CLUSTER ∪ {k_{l−1}, ..., k_l − 1}
10         i ← i − 1
11         j ← k_l
12     end
13 CLUSTER ← CLUSTER ∪ {k_l, ..., n}
```

In the example of Figure 4.4, the call LPART-CLUSTER(1,6,3) first refers to the entry b_{51} which contains $k_1 = 2$. Then, the subsequent value of k is obtained; $k_2 = 3$. Hence, an optimal partition of 3 clusters is $\{\{1\}, \{2\}, \{3, 4, 5, 6\}\}$.

Once we have computed the cumulative cost matrix in $O(n^2)$, the procedure LPART computes the optimal cost in $O(mn^2)$. The other steps in the loop of lines 6-12 require $O(m)$. Hence, the total time complexity is $O(mn^2)$.

4.2.5 Constraint on the Cluster Size

Let us now consider the constraint imposed on the size of a cluster. Denote s as the maximum size of a cluster allowed in the process of partitioning such that $ms \geq n$. The following pseudocode MLPART$(1, n, m, s)$ is obtained by simply adjusting the k values in line 9 of the LPART$(1, n, m)$ procedure. The lowest k value is changed from $(j+1)$ to $\max(j+1, n-(p-1)s+1)$ and the highest is changed from $(n-p+2)$ to $\min(n-p+2, j+s)$ in order to generate the valid k values for the constraint.

```
MLPART(1, n, m, s)
if ms < n
    return
for j ← 1 to n
    b_jj ← 0
for p ← 2 to m
    for j ← n − p + 1 downto 1
        begin
            b_jj ← ∞
            low ← max(j + 1, n − (p − 1)s + 1)
```

	1	2	3	4	5	6
1	0	118	198	220	199	129
2	0	0	98	139	142	107
3	0	0	0	71	102	87
4	0	0	0	0	49	59
5	0	0	0	0	0	27
6	0	0	0	0	0	0

p = 1

	1	2	3	4	5	6
1	220	118	198	220	199	129
2	0	139	98	139	142	107
3	0	0	71	71	102	87
4	0	0	0	49	49	59
5	0	0	0	0	27	27
6	4	4	4	5	6	0

p = 2

	1	2	3	4	5	6
1	257	118	198	220	199	129
2	0	169	98	139	142	107
3	0	0	120	71	102	87
4	0	0	0	76	49	59
5	2	3/5	4	5	27	27
6	4	4	4	5	6	0

p = 3

	1	2	3	4	5	6
1	287	118	198	220	199	129
2	0	215	98	139	142	107
3	0	0	147	71	102	87
4	2	4	4	76	49	59
5	2	3/5	4	5	27	27
6	4	4	4	5	6	0

p = 4

	1	2	3	4	5	6
1	333	118	198	220	199	129
2	0	245	98	139	142	107
3	2	3	147	71	102	87
4	2	4	4	76	49	59
5	2	3/5	4	5	27	27
6	4	4	4	5	6	0

p = 5

	1	2	3	4	5	6
1	363	118	198	220	199	129
2	2	245	98	139	142	107
3	2	3	147	71	102	87
4	2	4	4	76	49	59
5	2	3/5	4	5	27	27
6	4	4	4	5	6	0

p = 6

Figure 4.5. Computation of the MLPART(1,6,p,3) Problem for $1 \leq p \leq 6$

```
            high ← min(n − p + 2, j + s)
            if low ≤ high
                for k ← low to high
                    begin
                        sum ← b_jk + b_kk
                        if sum < b_jj
                            begin
                                b_jj ← sum
                                b_(n−p+2)j ← k
                            end
                    end
        end
```

Figure 4.5 illustrates the MLPART(1,6,p,3) problem for $1 \leq p \leq 6$. The indices of the lightly shaded area for a particular p represent the valid k values which are used for the next step of $(p+1)$, satisfying that the size of any cluster in a resulting optimal partition is equal to or less than 3. For example, consider the first row of the table for $p = 1$ which is used to compute $\phi_1(2)$ recorded in the b_{11} element of the table for $p = 2$. Unlike in the example of Figure 4.4, only the b_{14} element is used to

compute $\phi_1(2)$ because the consideration of other elements results in a cluster with the size greater than 3. It means that the *low* variable is set to $n - (p-1)s + 1 = 6 - (2-1) \times 3 + 1 = 4$ and the *high* variable is set to $j + s = 1 + 3 = 4$. Thus, $\phi_1(2) = b_{14} + b_{44} = 220 + 0 = 220$. Hence, the optimal partition is $\{\{1,2,3\},\{4,5,6\}\}$ if $p = 2$. Note that for each $p \geq 4$, the optimal cost $\phi_1(p)$ and partition Π are the same as the example of Figure 4.4 because the constraint does not limit the range of the k values. It always holds when $n - (p-1) \leq s$.

Given the maximum size of a cluster s and an integer K which is the maximum communication cost allowed for the entire system, the MLPART procedure can also be used to generate all possible optimal partitions for a sequence of n nodes. In the example of Figure 4.5, consider the case when $n = 6$, $s = 3$, and $K = 500$. Because the MLPART procedure optimizes the relative cost, we first compute the relative cost $K' = K - Cost_{constant} = 500 - 237 = 263$. Then, the MLPART procedure computes $\phi_1(p)$ for $2 \leq p \leq n$ until $\phi_1(p) \geq K'$. Because $\phi_1(2) = 220$ and $\phi_1(3) = 257$ are less than K', the possible optimal partitions are $\{\{1,2,3\},\{4,5,6\}\}$ if $p = 2$ and $\{\{1\},\{2,3\},\{4,5,6\}\}$ if $p = 3$.

4.3 Chapter Summary

In highway cellular systems, a linear array of n heterogeneous base stations may generate multiple types of traffic among themselves. We have considered the problem of finding a set of disjoint clusters to cover n base stations such that it minimizes the total communication cost for the entire system. The total communication cost consists of the costs incurred by two different types of traffic, mobility management data and user data. For each type of traffic, the cost of intra-cluster communication is usually lower than that of inter-cluster communication. The optimal partitioning problem is solvable in polynomial time of $O(mn^2)$ by dynamic programming for

an arbitrary number of clusters and size of a cluster, where m is the number of clusters in a partition. In addition, the algorithm finds all valid partitions in the same polynomial time if given a constraint on the cluster size and the total allowable communication cost for the entire system.

The optimal deployment of virtual cells in a hexagonal mesh arrangement of n base stations is described in the following chapter.

CHAPTER 5
MULTIWAY PARTITIONING IN HEXAGONAL CELLULAR SYSTEMS

Given a hexagonal mesh of base stations in cellular systems we consider the problem of finding a cover of disjoint clusters of base stations which generate multiple types of traffic among themselves. The problem differs from general graph partitioning problems in that it considers not only communication costs but also the underlying topology among base stations. The objective is to minimize the total communication cost for the entire system where inter-cluster communication is more expensive than intra-cluster communication for each type of traffic. The problem is transformed into the dual based on a topology matrix and a relative cost matrix. We develop several heuristics for the dual. These heuristics produce optimal partitions with respect to the initial partition, based on the techniques of moving or interchanging the boundary nodes between adjacent clusters. The heuristics are compared and shown to behave quite well through experimental tests and analysis.

5.1 Problem Formalization

5.1.1 Labeling Scheme

The physical deployment of n base stations is represented by a planar graph of a hexagonal mesh of n base stations, where the vertices represent base stations and the edges represent the adjacency of base stations. The vertices on the exterior face of the graph are level 1 vertices and the vertices on the exterior face of the subgraph induced by removing level 1 vertices are level 2 vertices, and so on. A planar graph

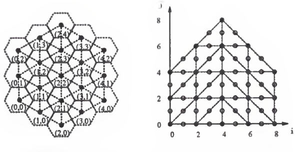


Figure 5.1. Labeling of the Physical Topology for H_3

is *s-outerplanar* if it has no vertices of level greater than s. Denote H_s as an s-outerplanar graph of a hexagonal mesh. It is then known that the number of vertices in an H_s is $n = 3s^2 - 3s + 1$ and the number of columns in each of three directions is $d = 2s - 1$. Before formalizing the problem, it is necessary to introduce a labeling scheme to describe the problem mathematically.

Starting from the left-most column of an H_s, each column is indexed from 0 through $d-1$ in sequence. Then the bottom vertices of every column constitute row 0, the next vertices of every column row 1, and so forth. Once the column index c and the row index r of a vertex is determined, the vertex is labeled (i, j) such that $i = 2c$ and $j = 2r$. In this two-dimensional coordinate system of an H_s, a point (i, j) for $0 \leq i, j \leq 2(d-1)$ can represent either a vertex v_{ij} if i and j are even or an edge e_{ij} otherwise. Figure 5.1 illustrates the labeling of the physical topology for an H_3 of $n = 19$ nodes.

If given a vertex v_{ij}, at most six edges, each leading to an adjacent vertex, are directly identified by the labeling scheme as follows:

- $e_{i(j+1)}$, $e_{(i+1)(j+1)}$, $e_{(i+1)j}$, $e_{i(j-1)}$, $e_{(i-1)(j-1)}$, and $e_{(i-1)j}$ if $i < d-1$;
- $e_{i(j+1)}$, $e_{(i+1)j}$, $e_{(i+1)(j-1)}$, $e_{i(j-1)}$, $e_{(i-1)(j-1)}$, and $e_{(i-1)j}$ if $i = d-1$;
- $e_{i(j+1)}$, $e_{(i+1)j}$, $e_{(i+1)(j-1)}$, $e_{i(j-1)}$, $e_{(i-1)j}$, and $e_{(i-1)(j+1)}$ if $i > d-1$.

On the other hand, if given an edge e_{ij}, the two vertices connected by the edge are directly identified by the labeling scheme as follows:

- $v_{(i-1)j}$ and $v_{(i+1)j}$ if i is odd and j is even;
- $v_{i(j-1)}$ and $v_{i(j+1)}$ if i is even and j is odd;
- $v_{(i-1)(j-1)}$ and $v_{(i+1)(j+1)}$ if both i and j are odd and $i < d$;
- $v_{(i+1)(j-1)}$ and $v_{(i-1)(j+1)}$ if both i and j are odd and $i > d$.

In the example of Figure 5.1(b), a vertex v_{24} leads to the six edges e_{13},e_{14},e_{23},e_{25}, e_{34}, and e_{35}, which are connected to the neighboring vertices, v_{02},v_{04},v_{22},v_{26},v_{44}, and v_{46}, respectively.

5.1.2 Topology Matrix

To handle the underlying topology constraint on clustering, we construct a topology matrix $\mathbf{T} = (t_{ij})$ for $0 \leq i,j \leq 2(d-1)$, where t_{ij} corresponds to either a vertex v_{ij} if i and j are even or an edge e_{ij} otherwise in the labeling scheme. If v_{ij} currently belongs to cluster P_x, the element t_{ij} is set to x. If the two vertices connected by e_{ij} belong to clusters P_x and P_y, the element t_{ij} is set to xy for $x \leq y$. Figure 5.2(b) shows the topology matrix $\mathbf{T}$ derived from an initial partition of 3 clusters in Figure 5.2(a).

From the topology matrix $\mathbf{T}$, the indices of the boundary edges between a pair of adjacent clusters P_x and P_y can be represented by a list of edge elements $L_{xy} = \{e_{ij} | t_{ij} = xy, x < y\}$. Using the list L_{xy}, the boundary vertices between them can be directly identified by the labeling scheme itself. Denote $V_{\bar{x}y}$ and $V_{x\bar{y}}$ as the indices of the boundary vertices for P_x and P_y, respectively. In the example of Figure 5.2, $L_{12} = \{e_{03}, e_{13}, e_{23}, e_{33}, e_{43}\}$, $V_{\bar{1}2} = \{v_{04}, v_{24}, v_{44}\}$, and $V_{1\bar{2}} = \{v_{02}, v_{22}, v_{42}\}$.

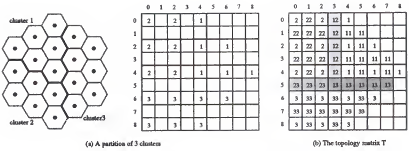


Figure 5.2. Topology Matrix for a Partition of H_3

5.1.3 Relative Cost Matrix

The communication among base stations is considered as a full mesh of point-to-point logical network to represent a possible communication between mobile hosts through different base stations. The communication network is described by a complete directed graph $G = (V, E)$, where $|V| = n$. The vertices of the graph represent base stations and the edges represent directional communication links between base stations. Each edge is assigned a move frequency by a function $f_m : V \times V \to R^+$ and a find frequency by $f_f : V \times V \to R^+$. Denote w_m^1 and w_m^2 as the weight of a move operation within a cluster and between clusters, respectively, and w_f^1 and w_f^2 as that of a find operation within a cluster and between clusters, respectively. Then we define a relative cost function $c : V \times V \to R^+$ for all $(v_{ij}, v_{kl}) \in V \times V$ as

$$c(v_{ij}, v_{kl}) = f_m(v_{ij}, v_{kl})(w_m^2 - w_m^1) + f_f(v_{ij}, v_{kl})(w_f^2 - w_f^1), \tag{5.1}$$

where $(w_m^2 - w_m^1)$ is the relative weight of a move operation and $(w_f^2 - w_f^1)$ is the relative weight of a find operation. The cost $c(v_{ij}, v_{kl})$ represents the total relative cost of $f_m(v_{ij}, v_{kl})$ move and $f_f(v_{ij}, v_{kl})$ find operations from v_{ij} to v_{kl} if they are in different clusters. Denote $c(v_{ij}, v_{kl})$ as $c_{ij;kl}$. The relative cost matrix $\mathbf{C} = (c_{ij;kl})$ for $0 \leq i, j, k, l \leq 2(d-1)$ is derived by the equation (5.1) from the move and find frequencies among n vertices.

5.1.4 Dual Problem

Let $\Pi = \{P_1, \ldots, P_m\}$ be a partition of m clusters of contiguous vertices such that $P_x \cap P_y = \emptyset$ and $\cup_x P_x = V$ for $x, y = 1, \ldots, m$ and $x \neq y$. The intra-cluster communication cost for all $(v_{ij}, v_{kl}) \in V \times V$ is

$$Cost_{intra} = \sum_{(v_{ij}, v_{kl}) \in V \times V} (f_m(v_{ij}, v_{kl}) w_m^1 + f_f(v_{ij}, v_{kl}) w_f^1)$$

and the relative inter-cluster communication cost of Π is

$$Rcost_{inter}(\Pi) = \sum_{x=1}^{m-1} \sum_{v_{ij} \in P_x, v_{kl} \in P_y, x<y} (c_{ij;kl} + c_{kl;ij}).$$

The total communication cost of Π is then

$$Cost_{total}(\Pi) = Cost_{intra} + Rcost_{inter}(\Pi).$$

Because $Cost_{intra}$ is constant independent of how to partition, it holds that Π minimizes $Cost_{total}$ iff Π minimizes $Rcost_{inter}$. Let us define the relative intra-cluster communication cost of Π as

$$Rcost_{intra}(\Pi) = \sum_{x=1}^{m} \sum_{v_{ij}, v_{kl} \in P_x} (c_{ij;kl} + c_{kl;ij}).$$

Because the sum of $Rcost_{intra}$ and $Rcost_{inter}$ is constant, our task of finding an optimal partition Π which minimizes $Rcost_{inter}$ is equivalent to that of finding an optimal partition Π which maximizes $Rcost_{intra}$.

Hence, the dual problem is: given the topology matrix $\mathbf{T} = (t_{ij})$ and the relative cost matrix $\mathbf{C} = (c_{ij;kl})$ for a system of n hexagonal cells, find a cover of m disjoint clusters of contiguous base stations $\Pi = \{P_1, \ldots, P_m\}$, so as to maximize $Rcost_{intra}(\Pi)$.

5.2 Heuristics

We consider heuristics for the problem: starting with an arbitrary partition Π of m clusters, we try to increase the initial relative intra-communication cost $Rcost_{intra}(\Pi)$

by repeated applications of a two-way optimization procedure to pairs of adjacent clusters. In the two-way optimization procedure, we try to increase the sum of the initial relative intra-communication cost of each cluster by moving or interchanging boundary nodes until no further improvement is possible. The resulting pair of adjacent clusters are then pairwise optimal with respect to the initial partition. Because the two-way optimality for a pair of adjacent clusters may affect that for another pair of adjacent clusters, more than one pass through all pairs of adjacent clusters may be required. This process can be repeated with another initial partition Π of m clusters, and so on, so as to obtain as many locally maximum partitions as we desire. Then, one of the resulting partitions has a fairly high probability of being a globally maximum partition.

5.2.1 Interchanging or Moving Boundary Nodes

Consider a pair of adjacent clusters P_x and P_y in the two-way optimization procedure. The interchange of a pair of boundary nodes, $v_{ij} \in V_{\bar{x}y}$ and $v_{kl} \in V_{x\bar{y}}$, is said to be *feasible* if the resulting clusters preserve the underlying topology constraint. In other words, all nodes in P_x (P_y) adjacent to v_{ij} (v_{kl}) must be connected in their physical topology before the interchange and v_{ij} (v_{kl}) must be connected to at least one of the nodes in P_y (P_x) after the interchange.

To determine feasible pairs of interchanging nodes, we use the topology matrix $\mathbf{T}$ where an element t_{ij} corresponds to a node v_{ij} or an edge e_{ij}. Given a node $v_{ij} \in V_{\bar{x}y}$, at most six edges adjacent to the node can be directly identified by the labeling scheme itself. Denote $Adj(v_{ij})$ as an ordered list of those edges which are sequentially arranged in a circular fashion. Then it is said that the node v_{ij} can be moved into P_y for the interchange when the following two conditions are satisfied:

	0	1	2	3	4	5	6	7	8
0	2	22	2	12	1				
1	22	12	12	12	11	11			
2	2	12	1	11	1	11	1		
3	22	22	12	12	12	11	11	11	
4	2	22	2	22	2	12	1	11	1
5	23	23	23	23	23	13	13	13	
6	3	33	3	33	3	33	3		
7	33	33	33	33	33	33			
8	3	33	3	33	3				

(a) after the interchange of (2,2) and (4,4.)

	0	1	2	3	4	5	6	7	8
0	2	22	2	12	1				
1	22	22	22	12	11	11			
2	2	22	2	12	1	11	1		
3	22	12	12	22	12	11	11	11	
4	2	12	1	12	2	12	1	11	1
5	23	13	13	23	23	13	13	13	
6	3	33	3	33	3	33	3		
7	33	33	33	33	33	33			
8	3	33	3	33	3				

(b) after the interchange of (4,2) and (4,4)

Figure 5.3. Interchanging Boundary Node Elements

1. There is only one continuous subsequence of edges which are equal to xx in $Adj(v_{ij})$.

2. There is at least one edge in $Adj(v_{ij})$, which is equal to xy and does not lead to the node $v_{kl} \in V_{x\bar{y}}$.

Assume that initially all nodes of each cluster are connected in their physical topology. Condition 1 implies that the remaining nodes of P_x after removing v_{ij} still preserve the connectivity in their physical topology. Condition 2 implies that the removed node v_{ij} is also connected to its adjacent cluster P_y whose nodes are already connected in their physical topology. In the same way, the node v_{kl} must also satisfy the above two conditions so that it can be moved into P_x for the interchange. Therefore, the interchange of a pair of boundary nodes $v_{ij} \in V_{\bar{x}y}$ and $v_{kl} \in V_{x\bar{y}}$ is feasible iff both v_{ij} and v_{kl} satisfy Condition 1 before the interchange and Condition 2 after the interchange.

In the example of Figure 5.2(b), we consider interchanging $v_{22} \in P_2$ and $v_{44} \in P_1$. For node v_{22}, since the only one continuous subsequence $\{e_{12}, e_{11}, e_{21}, e_{32}\}$ in $Adj(v_{22}) = \{e_{23}, e_{12}, e_{11}, e_{21}, e_{32}, e_{33}\}$ is equal to 22, Condition 1 holds. In addition, since $e_{23} = 12$ and it does not lead to v_{44}, Condition 2 also holds. Thus, v_{22} can be

moved to P_1 for the interchange. At the same time, for node v_{44}, since the only one continuous subsequence $\{e_{45}, e_{34}\}$ in $Adj(v_{44}) = \{e_{34}, e_{33}, e_{43}, e_{53}, e_{54}, e_{45}\}$ is equal to 11, Condition 1 holds. In addition, since $e_{43} = 12$ and it does not lead to v_{22}, Condition 2 also holds. Thus, v_{44} can be moved to P_2 for the interchange. Hence, v_{22} and v_{44} are a feasible pair of interchanging nodes between P_1 and P_2. Figure 5.3(a) shows the resulting topology matrix $\mathbf{T}$ after the interchange. On the other hand, the nodes $v_{42} \in P_2$ and $v_{44} \in P_1$ cannot be interchanged, as depicted in Figure 5.3(b). The node v_{42} is isolated after the interchange due to the fact that no edge except e_{43} adjacent to v_{42} is equal to 12 in Figure 5.2(b).

It should be noted that the move of a boundary node $v_{ij} \in V_{\overline{x}y}$ into P_y is rather simple than the interchange because we need to check that only all nodes in P_x adjacent to v_{ij} are connected in their physical topology before the move.

5.2.2 Computing Gains

When interchanging boundary nodes. Assume that a pair of boundary nodes, $v_{ij} \in V_{\overline{x}y}$ and $v_{kl} \in V_{x\overline{y}}$, is feasible for the interchange. Define the internal cost of node v_{ij} with respect to P_x for the interchange to be

$$I_e(v_{ij}) = \sum_{v_{i'j'} \in P_x, v_{ij} \neq v_{i'j'}} (c_{ij;i'j'} + c_{i'j';ij}),$$

and the external cost of node v_{ij} with respect to P_y for the interchange to be

$$E_e(v_{ij}) = \sum_{v_{k'l'} \in P_y, v_{kl} \neq v_{k'l'}} (c_{ij;k'l'} + c_{k'l';ij}).$$

Similarly, we define $I_e(v_{kl})$ and $E_e(v_{kl})$ for the boundary node v_{kl}. If v_{ij} and v_{kl} are interchanged, then the gain g of the increase in cost is given by

$$g = (E_e(v_{ij}) + E_e(v_{kl})) - (I_e(v_{ij}) + I_e(v_{kl})).$$

When moving boundary nodes. Assume that a node $v_{ij} \in V_{\bar{x}y}$ is feasible for the move. Define the internal cost of node v_{ij} with respect to P_x for the move to be

$$I_m(v_{ij}) = \sum_{v_{i'j'} \in P_x, v_{ij} \neq v_{i'j'}} (c_{ij;i'j'} + c_{i'j';ij}),$$

and the external cost of node v_{ij} with respect to P_y for the move to be

$$E_m(v_{ij}) = \sum_{v_{k'l'} \in P_y} (c_{ij;k'l'} + c_{k'l';ij}).$$

If $v_{ij} \in P_x$ is moved into P_y, then the gain g of the increase in cost is given by

$$g = E_m(v_{kl}) - I_m(v_{ij}).$$

5.2.3 Heuristic.1

Given a pair of adjacent clusters P_x and P_y, Heuristic.1 interchanges only one feasible pair of boundary nodes with a positive maximum gain g between the two clusters. This process is repeated with an updated boundary until no feasible pair produces a positive gain. Let k be the number of feasible pairs interchanged. Then the the total gain with respect to the sum of the initial costs for the two clusters is $G_{xy} = \sum_{i=1}^{k} g_i$. Note that a pair of nodes interchanged in the previous step is not interchanged again at the next step because the positive maximum gain in the previous step becomes the minimum gain with the same negative value in the next step.

Heuristic.1(Π)
for every pair of adjacent clusters P_x and P_y in Π
    begin
        Determine the boundary node lists $V_{\bar{x}y}$ and $V_{x\bar{y}}$
        forever
            begin
                Determine a set of feasible pairs
                if there is no feasible pair
                    break

Compute gains for all feasible pair
if there is no feasible pair with a positive gain
break
Choose a feasible pair (v_{ij}, v_{kl}) with the maximum gain
Interchange v_{ij} and v_{kl}
Update the boundary node lists $V_{\overline{x}y}$ and $V_{x\overline{y}}$
end
end

5.2.4 Heuristic.2

Once a set of feasible pairs of boundary nodes is determined from the boundary node lists $V_{\overline{x}y}$ and $V_{x\overline{y}}$, Heuristic.2 interchanges all feasible pairs with positive gains before updating the boundary node lists $V_{\overline{x}y}$ and $V_{x\overline{y}}$. Initially, all feasible pairs are unmarked. A feasible pair with the maximum positive gain is first interchanged and then all remaining unmarked feasible pairs which involve the boundary nodes of the interchanged pair are marked so that in the next step they cannot be considered again. After marking, the gains for all unmarked feasible pairs are computed again and then an unmarked feasible pair with the largest positive gain is next interchanged.

Heuristic.2(Π)
for every pair of adjacent clusters P_x and P_y in Π
begin
Determine the boundary node lists $V_{\overline{x}y}$ and $V_{x\overline{y}}$
forever
begin
Determine and unmark a set of feasible pairs
if there is no feasible pair
break
Compute gains for all feasible pairs
while there are unmarked feasible pairs with positive gains
begin
Choose a pair (v_{ij}, v_{kl}) with the largest gain
Interchange v_{ij} and v_{kl}
Mark all unmarked feasible pairs involving v_{ij} or v_{kl}
Compute gains for all unmarked feasible pairs
end

```
17              Update the boundary node lists V_{x̄y} and V_{xȳ}
18          end
19      end
```

5.2.5 Heuristic.3

Heuristic.3 is based on the observation that infeasible pairs of boundary nodes might become feasible pairs after interchanging feasible pairs. For example, consider a pair of adjacent clusters P_1 and P_2 in the topology matrix of H_3, where $P_1 = \{v_{00}, v_{02}, v_{04}\}$ and $p_2 = \{v_{20}, v_{22}, v_{24}, v_{26}\}$. Then $V_{\bar{1}2} = P_1$ and $V_{1\bar{2}} = P_2$. A pair of boundary nodes (v_{02}, v_{22}) is infeasible for the interchange because v_{02} and v_{22} violate Condition 1. However, if a feasible pair (v_{04}, v_{20}) is interchanged, the infeasible pair (v_{02}, v_{22}) becomes a feasible pair for the next interchange.

The algorithm of Heuristic.3 is obtained by slightly modifying that of Heuristic.2. Before performing line 15 in Heuristic.2, infeasible pairs of boundary nodes which become feasible pairs due to the interchange of nodes v_{ij} and v_{kl} in line 13 are added to the set of unmarked feasible pairs.

5.2.6 Heuristic.4

While the previous three heuristics interchange boundary nodes between a pair of adjacent clusters, Heuristic.4 moves a boundary node in a cluster into the other cluster between a pair of adjacent clusters. Since the move of a boundary node between a pair of adjacent clusters changes their cluster sizes, the constraint on the cluster size is given by the minimum and maximum cluster sizes.

Given a pair of adjacent clusters, Heuristic.4 determines a feasible boundary node with a maximum positive gain for each cluster and compares them to determine a boundary node to be moved. Once the boundary node to be moved is determined, it is removed from its cluster and added to the other cluster as long as its move does

not violate the constraint on the cluster size. This process is repeated until there is no feasible boundary node for the move or one of the clusters has the minimum or maximum cluster size.

Heuristic.4(Π)

```
for every pair of adjacent clusters $P_x$ and $P_y$ in Π
    begin
        Determine the boundary node lists $V_{\bar{x}y}$ and $V_{x\bar{y}}$
        while the constraint on the cluster size is satisfied
            begin
                Determine a set of feasible nodes
                if there is no feasible node
                    break
                Compute gains for all feasible nodes
                if there is no feasible node with a positive gain
                    break
                Choose a feasible node $v_{ij}$ with the maximum gain
                Move $v_{ij}$ into the other cluster
                Update the boundary node lists $V_{\bar{x}y}$ and $V_{x\bar{y}}$
            end
    end
```

5.2.7 Heuristic.5, Heuristic.6, and Heuristic.7

Heuristic.5, Heuristic.6, and Heuristic.7 are based on a repeated application of two-phase optimization. In the first phase, the heuristics use Heuristic.1, Heuristic.2, and Heuristic.3 for interchanging the boundary nodes between adjacent clusters, respectively. Their second phase uses Heuristic.4 for moving the boundary nodes between adjacent clusters to improve the gains obtained by their first phase if possible. The improvement by their second phase implies the change of the boundary nodes and the possibility that their first phase further improves their gains. Thus, the two-phase optimization is repeatedly applied until no improvement is possible by their second phase. The basic idea of the heuristics is that changing the cluster size by their second phase might overcome the limited improvement due to preserving the cluster sizes of the initial partition by their first phase.

5.2.8 Heuristic.8, Heuristic.9, and Heuristic.10

Heuristic.8, Heuristic.9, and Heuristic.10 are also based on a repeated application of two-phase optimization. However, in the first phase, the heuristics use Heuristic.4 for moving the boundary nodes between adjacent clusters. Their second phase uses respectively Heuristic.1, Heuristic.2, and Heuristic.3 for interchanging the boundary nodes between adjacent clusters to improve the gains obtained by their first phase if possible. The heuristics are derived from the fact that the limited gain of increase in cost due to the constraint on the cluster size in their first phase might be further improved by interchanging the boundary nodes without violating the constraint on the cluster size.

5.3 Experimental Testing and Analysis

We obtain the initial partition for experimental testing of the heuristics by two different methods: *random* and *centering*. The random partition is based on the topology matrix because it is only concerned with the geographical arrangement of base stations and the cluster size constraint. On the other hand, the centering partition is based on both the topology and relative cost matrices because it further considers the traffic pattern among base stations.

The centering partition of m clusters is achieved by a two-phase algorithm. In the first phase, the m center nodes on which traffics are concentrated are selected by using the relative cost matrix. Each center node forms an initial intermediate cluster. For every intermediate cluster, the second phase identifies the adjacent nodes of the cluster which are not involved in other clusters by using the topology matrix and chooses one of them by using the relative cost matrix, which produces a maximum increase in cost when it is involved in the cluster. Then the maximum node is added

Table 5.1. Number of Times in 100 Trials a Heuristic Produced a Solution with Maximal Total Cost with Respect to Other Heuristics When $m = 3$ and $1/2 \times \lceil n/m \rceil \leq |P_i| \leq 2 \times \lceil n/m \rceil$

	Random				Centering				
Heuristics	n=19	n=37	n=61	n=91	n=19	n=37	n=61	n=91	Total
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	17	6	2	1	17	8	1	2	54
5	84	48	47	37	88	76	56	50	486
6	84	51	43	37	86	80	70	45	496
7	85	51	47	33	85	71	70	45	487
8	70	59	39	41	76	62	57	49	453
9	70	62	41	40	76	63	60	52	464
10	70	61	43	38	75	62	61	52	462

to the intermediate cluster. This process is repeated until all nodes are contained one of the m clusters.

In experimental testing, the 100 instances of the relative cost matrix were randomly generated for each of several values of n, where $n = 19, 37, 61$, and 91 for H_3, H_4, H_5, and H_6, respectively. For each value of n, Heuristic.1 through Heuristic.10 were extensively tested on the 100 cost matrix instances for each of the four cases which are the combinations of the two methods for obtaining the initial partition, random or centering, and the number of clusters $m = 3$ or 4. The constraint on the cluster size is given by $1/2 \times \lceil n/m \rceil \leq |P_i| \leq 2 \times \lceil n/m \rceil$.

Table 5.1 and Table 5.2 present the number of times a heuristic produces a solution that is maximal with respect to the solutions produced by the other heuristics when $m = 3$ and 4, respectively. The tables show that Heuristic.5 through Heuristic.10 which are the combinations of the techniques of interchanging or moving boundary nodes greatly outperform the other heuristics using only one of the techniques. This

Table 5.2. Number of Times in 100 Trials a Heuristic Produced a Solution with Maximal Total Cost with Respect to Other Heuristics When $m = 4$ and $1/2 \times \lceil n/m \rceil \leq |P_i| \leq 2 \times \lceil n/m \rceil$

Heuristics	Random				Centering				Total
	n=19	n=37	n=61	n=91	n=19	n=37	n=61	n=91	
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	8	2	2	2	7	2	2	1	26
5	43	46	41	33	73	71	58	52	417
6	43	45	43	42	73	72	61	52	431
7	64	53	56	45	84	71	63	60	496
8	68	59	34	27	80	74	66	41	449
9	68	59	36	27	80	75	71	45	461
10	68	61	37	29	80	76	66	43	460

is due to the fact that although the cost between a pair of adjacent clusters cannot be improved by interchanging their boundary nodes, moving their boundary nodes might further improve the cost, which in turn changes the boundary between the adjacent clusters for possible interchanging at the next step, and vice versa. The tables also show that as the number of nodes n increases, the probability that a heuristic produces a maximal solution becomes lower.

It is interesting that Heuristic.1 through Heuristic.3 do not produce maximal solutions at all even though Heuristic.4 produces a small number of solutions being maximal. This is because interchanging boundary nodes does not change the number of nodes for each cluster in the initial partition and so less flexible than moving boundary nodes. Thus, unlike general graph partitioning problems, the algorithm which is only based on one of the techniques of interchanging or moving boundary nodes is no longer useful for our problem which additionally considers the underlying topology.

Table 5.3. Maximum Difference in Total Cost From Maximal Solution When $m = 3$ and $1/2 \times \lceil n/m \rceil \leq |P_i| \leq 2 \times \lceil n/m \rceil$

Heuristics	Random				Centering			
	n=19	n=37	n=61	n=91	n=19	n=37	n=61	n=91
1	44.2	36.4	34.6	31.4	42.8	37.1	35.0	34.7
2	44.2	36.4	34.6	31.2	42.8	37.1	35.0	34.7
3	44.2	36.4	33.1	31.0	42.8	37.1	35.0	34.7
4	9.3	6.3	5.6	4.3	11.2	17.7	5.7	23.4
5	8.7	28.1	25.9	13.5	27.3	3.2	23.5	23.1
6	8.7	28.1	25.9	18.9	27.3	3.2	23.5	23.1
7	8.7	26.3	25.9	18.9	27.3	4.5	2.4	1.6
8	7.2	23.3	25.6	23.9	3.9	3.7	2.7	2.0
9	7.2	23.3	25.6	23.9	3.9	3.7	2.7	2.0
10	7.2	23.3	3.1	23.9	3.9	3.7	2.7	2.0

Table 5.4. Maximum Difference in Total Cost From Maximal Solution When $m = 4$ and $1/2 \times \lceil n/m \rceil \leq |P_i| \leq 2 \times \lceil n/m \rceil$

Heuristics	Random				Centering			
	n=19	n=37	n=61	n=91	n=19	n=37	n=61	n=91
1	40.2	33.1	29.5	26.3	40.6	34.6	31.5	29.9
2	40.2	33.1	29.7	26.1	40.6	34.6	31.5	29.9
3	40.2	33.1	29.7	26.0	40.6	34.6	31.5	29.9
4	16.0	8.2	5.5	5.0	23.3	17.9	9.4	5.4
5	11.2	5.2	10.3	4.5	14.0	17.0	12.8	3.7
6	11.2	5.2	10.3	3.5	14.0	17.0	12.8	3.7
7	9.8	5.2	2.7	4.1	6.8	4.5	3.3	2.1
8	13.1	5.4	3.9	4.3	6.8	4.7	3.6	3.9
9	13.1	5.4	3.9	4.3	6.8	4.7	3.6	3.9
10	13.1	5.4	3.9	4.3	6.8	4.7	3.6	3.9

Table 5.5. Average Difference in Total Cost From Maximal Solution When $m = 3$ and $1/2 \times \lceil n/m \rceil \leq |P_i| \leq 2 \times \lceil n/m \rceil$

	Random				Centering			
Heuristics	n=19	n=37	n=61	n=91	n=19	n=37	n=61	n=91
1	38.2	33.6	31.0	29.2	35.1	32.3	31.7	31.9
2	38.2	33.6	31.0	29.2	35.1	32.2	31.7	31.9
3	38.2	33.6	31.0	29.1	35.1	32.2	31.7	31.9
4	3.0	2.3	2.3	1.7	3.1	2.4	1.6	1.5
5	0.4	1.6	1.0	0.8	0.7	0.2	0.7	0.5
6	0.4	1.3	1.4	0.8	0.7	0.2	0.6	0.5
7	0.4	1.0	1.3	1.1	0.6	0.4	0.2	0.3
8	0.9	0.7	0.9	0.9	0.4	0.4	0.3	0.3
9	0.9	0.6	0.9	0.9	0.4	0.4	0.3	0.3
10	0.9	0.6	0.6	0.7	0.4	0.4	0.3	0.3

Table 5.6. Average Difference in Total Cost From Maximal Solution When $m = 4$ and $1/2 \times \lceil n/m \rceil \leq |P_i| \leq 2 \times \lceil n/m \rceil$

	Random				Centering			
Heuristics	n=19	n=37	n=61	n=91	n=19	n=37	n=61	n=91
1	31.3	29.2	27.5	24.2	31.8	29.3	26.1	25.4
2	31.3	29.2	27.5	24.2	31.8	29.3	26.1	25.5
3	31.3	29.2	27.4	24.2	31.8	29.3	26.1	25.4
4	5.9	3.3	2.8	2.1	5.2	3.2	2.2	2.0
5	2.3	0.7	0.7	0.7	1.1	0.7	0.5	0.4
6	2.3	0.6	0.7	0.5	1.1	0.7	0.5	0.4
7	0.9	0.5	0.4	0.4	0.5	0.5	0.3	0.3
8	0.9	0.5	0.7	0.6	0.6	0.5	0.2	0.5
9	0.9	0.5	0.8	0.6	0.6	0.4	0.2	0.5
10	0.9	0.5	0.7	0.6	0.6	0.4	0.2	0.5

Table 5.3 and Table 5.4 present the percent of maximum difference in cost from the solution with the maximal cost, while Table 5.5 and Table 5.6 present the percent of average difference in cost from the solution with the maximal cost. The tables confirm the superiority of Heuristic.5 through Heuristic.10 with respect to the other hueristics. In general, the maximum differences for the centering partition of Heuristic.8 through Heuristic.10 are minimal with respect to the random partition of Heuristic.8 through Heuristic.10 and both the random and centering partitions of Heuristic.5 through Heuristic.7 with some exceptional test cases. The average differences for the centering partition of Heuristic.8 through Heuristic.10 are also minimal in general.

5.4 Chapter Summary

In cellular systems, a hexagonal mesh of n base stations may generate multiple types of traffic among themselves. We have considered the problem of finding a cover of disjoint clusters of base stations, so as to minimize the total communication cost for the entire system. The problem differs from general graph partitioning problems in that it additionally considers the underlying physical topology among base stations. We have developed several heuristics based on the combinations of the techniques of moving or interchanging the boundary nodes between adjacent clusters. These heuristics produce optimal partitions with respect to the initial partition obtained randomly or by centering. The heuristics are compared and shown to behave quite well through experimental testing and analysis.

CHAPTER 6
A PERFORMANCE ANALYSIS OF THE VIRTUAL CELL SYSTEM

In this chapter, we deal with the performance analysis of the virtual cell system. Once an optimal partition of m disjoint clusters is obtained by the algorithms in Chapter 4 and Chapter 5, the virtual cell system is deployed as shown in Figure 3.3 where each cluster corresponds to a virtual cell. Virtual cell i, where $1 \leq i \leq m$, is implemented by interconnecting the base stations of the ith cluster and an ARP/Location server by a base station network. These m virtual cells are interconnected by the backbone network. The virtual cell system is modeled as a BCMP open multiple class queueing network. Both the move and find frequencies among base stations and the topology of the virtual cell system are used to determine service transition probabilities in the queueing network model as well as the arrival rate for each type of messages. There are three types of messages entering or leaving the virtual cell system via base stations: the handoff message, the data message, and the address resolution message. The handoff messages are generated due to move operations and the data and address resolution messages are due to find operations. By solving traffic equations, the performance measures such as the network response time and protocol processing loads at network components are obtained.

6.1 Performance Model

We adopt a BCMP open multiple class queueing network to model the virtual cell system, as depicted in Figure 6.1. A virtual cell is modeled as a number of base station nodes, an ARP/Location server node, and a base station network node which

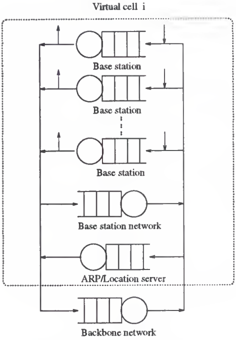
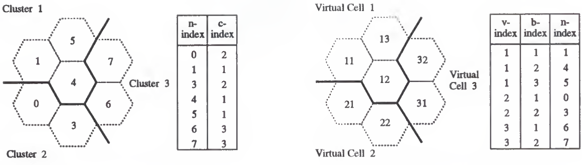


Figure 6.1. The Performance Model of the Virtual Cell System

captures traffic characteristics among physical cells in the same virtual cell. But, in order to capture traffic characteristics between virtual cells, a separate service node is used to model the backbone network. Messages from mobile hosts enter and leave the network model, only going through base station nodes in the virtual cell system.

The base station nodes of virtual cell i are sequentially indexed as $1, 2, \ldots, n_i$ in an arbitrary order, where n_i is the number of base stations in virtual cell i such that $\sum_{i=1}^{m} n_i = n$. Denote ij as service node j in virtual cell i. The network model with $N = n + 2m + 1$ nodes is defined as follows:

1. A base station node is labeled ij, where $1 \leq i \leq m$ and $1 \leq j \leq n_i$, and modeled as an FCFS type of service station with a fixed service rate μ_{ijr} for message class r. The messages in the queue are transmitted to its base station network node or leave the network model.



n-index	c-index
0	2
1	1
3	2
4	1
5	1
6	3
7	3

(a) The labeling scheme in the optimization algorithms

v-index	b-index	n-index
1	1	1
1	2	4
1	3	5
2	1	0
2	2	3
3	1	6
3	2	7

(b) The corresponding labeling in the network model

Figure 6.2. Labeling the Topology of the Virtual Cell System

2. A base station network node is labeled iB, where $1 \leq i \leq m$, and modeled as a PS type of service station with a fixed service rate μ_{iB_r} for message class r. The messages in the queue are transmitted to its base station nodes or ARP/Location server node.

3. An ARP/Location server node is labeled iS, where $1 \leq i \leq m$, and modeled as an FCFS type of service station with a fixed service rate μ_{iS_r} for message class r. The messages in the queue are transmitted to its base station network nodes or the backbone network node.

4. The backbone network node is labeled $B1$ and modeled as a PS type of service station with a fixed service rate μ_{B1_r} for message class r. The messages in the queue are transmitted to ARP/Location server nodes of virtual cells.

The analysis is based on the following assumptions:

1. Given a hexagonal mesh of base stations of size s, it is known that the number of base stations are $n = 3s^2 - 3s + 1$ and the number of columns in each three directions is $d = 2s - 1$. From left to right, each column is indexed as $0, 1, 2, \ldots, d-1$. Let i be the column index. Then the nodes of every column i

are sequentially labeled $i \times d, i \times d+1, i \times d+2, \ldots$, from bottom to top. Denote *nindex* and *cindex* as the node and cluster indices, respectively. An optimal partition produced by the algorithms [23, 24] can be represented by an index pair $(nindex, cindex)$ for every base station. Figure 6.2(a) depicts an optimal partition for a hexagonal mesh of size 2. Note that the node index reflects the underlying topology of the hexagonal mesh of base stations. Thus, given the node index of a base station, its neighboring base stations can be directly identified by the labeling scheme of the algorithms.

In addition to the node index, the network model of the virtual cell system also requires base stations to be logically indexed because a base station in a virtual cell can change into another virtual cell according to traffic patterns. Denote *vindex* and *bindex* as the virtual cell and base station indices, respectively. The virtual cell index is directly mapped to the cluster index and the base station index is logically assigned so that the base stations in a virtual cell is sequentially labeled $1, 2, \ldots$, in an arbitrary order. Figure 6.2(b) depicts the labeling of the network model which corresponds to an optimal partition in Figure 6.2(a). Thus, given base station xy in the network model, where x and y are the virtual cell and base station indices, respectively, its neighboring base stations are directly identified by the node index. Let $Adj(xy)$ be a set of base stations adjacent to xy. From $Adj(xy)$ we can identify which neighboring base station belongs to which virtual cell.

2. We use a flow-based mobility model [25] which assumes that mobile hosts are uniformly distributed in the area of a physical cell and the travel direction of each mobile host with respect to the border is uniformly distributed. If we define ρ_{xy} to be the density of the mobile hosts per km^2 at the area of base

station xy, v_{xy} to be the average speed in km/sec of a mobile host at the area of base station xy, and L to be the length of the perimeter of the area of a physical cell, then the average number of mobile hosts per sec leaving base station xy is given by $M_{xy} = \rho_{xy} v_{xy} L/\pi$. Denote $f_m(xy, x'y')$ as the move frequency from base station xy to one of its neighboring base station $x'y' \in Adj(xy)$. Then $f_m(xy, x'y') = M_{xy}/|Adj(xy)|$, where $|Adj(xy)|$ is the number of base stations adjacent to xy and set to 6 in the hexagonal arrangement of n base stations.

3. If we define b to be the data rate in $bits/sec$ of a wireless channel, l to be the average message length in bits, and E to be the in-call probability in Erlangs for a mobile host, then the non-blocking data arrival rate in $messages/sec$ at base station xy is given by $F_{xy} = \pi R^2 \rho_{xy} Eb/l$, where R is the radius in km of a physical cell. Denote $f_f(xy, x'y')$ as the find frequency from base station xy to base station $x'y'$. Then $f_f(xy, x'y')$ is a fraction of F_{xy}, such that $\sum_{x'y'} f_f(xy, x'y') = F_{xy}$.

4. The address resolution frequency is derived from the find frequency. If we define $F_{\overline{x}y}$ to be the arrival rate in $messages/sec$ of data messages at base station xy, which are destined to base stations in the same virtual cell x, c to be the average number of messages over a conversation from the source mobile host to the destination mobile host, and h_m to be the miss ratio of the address resolution table of a mobile host, then the arrival rate in $messages/sec$ of the address resolution messages at base station xy is given by $A_{xy} = F_{\overline{x}y} h_m / c$, where $F_{\overline{x}y} = \sum_{xy'} f_f(xy, xy')$. Denote $f_a(xy, xy')$ as the address resolution frequency from base station xy to base station xy'. Then $f_a(xy, xy')$ is a fraction of A_{xy}, such that $\sum_{xy'} f_a(xy, xy') = A_{xy}$.

It should be noted that the move frequency not only affects the arrival rate of the handoff messages but also the forwarding rate of the data and address resolution reply messages. For example, consider the move frequency $f_m(xy, xy')$. When a mobile host at base station xy moves into base station xy', it initiates a handoff request message to xy'. But a data message destined to the mobile host may be forwarded to base station xy' at base station xy during the handoff process. In the same way, the address resolution reply message for the mobile host from the ARP/Location server should be also forwarded to base station xy' at base station xy.

6.1.1 Multiple Class Traffic Model

There are three classes of messages entering the network: the handoff request message, the data message, and the address resolution request message. As each class of message traverses through the network, it not only requires different service requirements and different routing behavior, but changes its class. For example, the base station received a handoff request message from a mobile host sends the message to the previous base station of the mobile host. Then the previous base station multicasts the message in its virtual cell to update the distributed location information of the mobile host, and at the same time it sends a handoff response message to the new base station from which the mobile host receives a handoff confirmation message. Thus, as a message of the *handoff request class* progresses through the network, it is changed into a message of the *multicast class* and next into a message of the *handoff response class.*

After the handoff completes, a data message destined to the mobile host will be directly delivered to the new base station. However, during the handoff the data message will be first delivered to the previous base station, which in turn forwards it to the new base station. Thus, as a message of the *data class* progresses through the

network, it might be changed into a forwarding message which may be involved in a virtual cell or between virtual cells. The forwarding message involved in a virtual cell is referred to as the *intra-forwarding class* message, while that involved between virtual cells is referred to as the *inter-forwarding class* message. In the same way, as a message of the *address resolution request class* progresses through the network, it is changed into a message of the *address resolution reply class* at ARP/Location servers and possibly the *address resolution reply forwarding class* at base stations.

Hence, the set of message classes in the network can be partitioned into three subsets which contain the message classes related to handoff, data, and address resolution, respectively. Denote $z = 1, 2$, and 3 as the subsets of the message classes related to handoff, data, and address resolution, respectively. Even though any class r in a subset z can visit the entire set of nodes N, we cannot define a routing chain for each subset z because a class r message in a subset z may require different routing behavior due to the topology of the virtual cell system.

For example, consider a handoff request message from a mobile host which moves from an adjacent base station into base station 12 in the example of Figure 6.2(b). If the mobile host is from base station 11, the message will be directly delivered to its previous base station 11 via base station network $1B$. However, if the mobile host is from base station 32, the message will be delivered to its previous base station 32, going through base station network $1B$, ARP/Location server $1S$, backbone network $B1$, ARP/Location server $3S$, and finally base station network $3B$. Thus, in addition to the message class, the topology of the virtual cell system should be considered to determine the routing behavior of messages in the network.

In order to incorporate the topology of the virtual cell system into the queueing network model, it is necessary to identify which base station generates the message classes for each subset z. Thus, we define a routing chain for the message classes of

each subset z generated by each base station. Then there are $3n$ routing chains in the network, denoted as $E_{x,y,z}$, where $1 \leq x \leq m$, $1 \leq y \leq n_x$, and $z = 1, 2$, and 3. For each routing chain $E_{x,y,z}$, the service transition probabilities are defined by the set $\{p_{ijr;kls}\}$, which describes the probability that a class r message at node ij goes next to node kl as a class s message, where $r, s \in z$ and $ij, kl \in N$.

6.1.2 Arrival Process

We assume a Poisson state-independent arrival stream for each routing chain. Denote $\lambda_{x,y,z}$ as the arrival rate corresponding to a routing chain $E_{x,y,z}$. Then the arrival rate $\lambda_{x,y,z}$ is determined by the move, find, and address frequencies among n base stations.

1. A mobile host migrating from base station $x'y' \in Adj(xy)$ to base station xy initiates a handoff request message to base station xy. Thus, the arrival rate for the handoff request class of messages at base station xy can be represented by $\lambda_{x,y,1} = \sum_{x'y' \in Adj(xy)} f_m(x'y', xy)$.

2. A data message transmitted by a mobile host at base station xy needs a find operation to locate base station $x'y'$ to which the destination mobile host belongs. Thus, the arrival rate for the data class of messages at base station xy can be represented by $\lambda_{x,y,2} = \sum_{x'y'} f_f(xy, x'y')$.

3. The source mobile host performs address resolution operations only when it resides in its native virtual cell and the network address of the destination mobile host indicates the same native virtual cell. Thus, the arrival rate for the address resolution request class of messages at base station xy can be represented by $\lambda_{x,y,3} = \sum_{x'y', x'=x} f_a(xy, x'y')$.

6.1.3 Service Transition Matrix

A service transition matrix $P_{x,y,z} = [p_{ijr;kls}]$ is to be defined for each routing chain $E_{x,y,z}$. To determine service transition probabilities, it is necessary to consider not only the move, find, and address resolution frequencies, but also the topology of the virtual cell system. For the handoff message classes, base station xy may send and receive a handoff request class message and a handoff response class message to and from an adjacent base stations $st \in Adj(xy)$ on a different route in the network, depending on whether base station st is located in the same virtual cell or a different virtual cell. For the data message classes, in the same way, base station xy may deliver a data class message to base station $st \in Adj(xy)$ on a different route, depending on whether base station st is located in the same virtual cell or a different virtual cell. If the data class message arriving at base station st is to be forwarded to an adjacent base station $s't' \in Adj(st)$, the forwarding message may also take a different route in the network model depending on whether $s't'$ and st are in the same virtual cell or different virtual cells.

Handoff message classes. Consider the handoff request, multicast, and handoff response message classes originated from base station xy, i.e, a routing chain $E_{x,y,1}$. Define $\delta_{st} = f_m(st, xy)/\lambda_{x,y,1}$ to be a probability that a mobile host at base station $st \in Adj(xy)$ moves into base station xy. Denote xy' as a base station adjacent to xy in the same virtual cell x and $x'y'$ as a base station adjacent to xy in a different virtual cell x'. Then we can compute the following probabilities from the move frequency:

- The probability that a mobile host at base station xy' moves into base station xy is given by $\delta_{xy'} = f_m(xy', xy)/\lambda_{x,y,1}$.
- The probability that a mobile host in virtual cell x moves into base station xy is given by $\delta_x = \sum_{xy' \in Adj(xy)} \delta_{xy'}$.

- The probability that a mobile host at base station $x'y'$ moves into base station xy is given by $\delta_{x'y'} = f_m(x'y', xy)/\lambda_{x,y,1}$.
- The probability that a mobile host in a different virtual cell x' moves into base station xy is given by $\delta_{x'} = \sum_{x'y' \in Adj(xy)} \delta_{x'y'}$.

The service transition matrix $P_{x,y,1}$ for the handoff message classes originated from base station xy is given as follows:

- For the handoff message classes when a mobile host moves from base station $xy' \in Adj(xy)$ into base station xy, service transition probabilities are as follows:
  - $p_{xy1;xB1} = 1.0$ and $p_{xB1;xy'1} = \delta_{xy'}$ for the handoff request class,
  - $p_{xy'1;xB2} = 1.0$ and $p_{xB2;xy'2} = \delta_{xy'}/\delta_x$ for the multicast class, and
  - $p_{xy'2;xB3} = 1.0$ and $p_{xB3;xy3} = 1.0$ for the handoff response class.
- For the handoff message classes when a mobile host moves from base station $x'y' \in Adj(xy)$ into base station xy, service transition probabilities are as follows:
  - $p_{xy1;xB1} = 1.0$, $p_{xB1;xS1} = 1-\delta_x$, $p_{xS1;B11} = 1.0$, $p_{B11;x'S1} = \delta_{x'}/\sum_{x' \in Adj(xy)} \delta_{x'}$, $p_{x'S1;x'B1} = 1.0$, and $p_{x'B1;x'y'1} = \delta_{x'y'}/\delta_{x'}$ for the handoff request class,
  - $p_{x'y'1;x'B2} = 1.0$ and $p_{x'B2;x'y'2} = \delta_{x'y'}/\delta_{x'}$ for the multicast class, and
  - $p_{x'y'2;x'B3} = 1.0$, $p_{x'B3;x'S3} = 1.0$, $p_{x'S3;B13} = 1.0$, $p_{B13;xS3} = 1.0$, $p_{xS3;xB3} = 1.0$, and $p_{xB3;xy3} = 1.0$ for the handoff response class.

Data message classes. Consider the data, intra-forwarding, and inter-forwarding message classes originated from base station xy, i.e, a routing chain $E_{x,y,2}$. Define

$\alpha_{st} = f_f(xy, st)/\lambda_{x,y,2}$ to be a probability that a mobile host at base station xy sends a data message to base station st. Denote xy' as a base station in the same virtual cell x and $x'y'$ as a base station in a different virtual cell x'. Then we can calculate the following probabilities for the data message class from the find frequency:

- The probability that a data message from base station xy is destined to base station xy' is given by $\alpha_{xy'} = f_f(xy, xy')/\lambda_{x,y,2}$.
- The probability that a data message from base station xy is destined to the base stations in virtual cell x is given by $\alpha_x = \sum_{xy'} \alpha_{xy'}$.
- The probability that a data message from base station xy is destined to a base station $x'y'$ is given by $\alpha_{x'y'} = f_f(xy, x'y')/\lambda_{x,y,2}$.
- The probability that a data message from base station xy is destined to the base stations in a different virtual cell x' is given by $\alpha_{x'} = \sum_{x'y'} \alpha_{x'y'}$.

Let us now consider forwarding message classes in the network. Consider the case that the source mobile host at base station xy trys to send a data message to the destination mobile host which is moving from base station st into base station $s't' \in Adj(st)$. If the message transmission occurs after the handoff completes, the message should be directly sent to base station $s't'$. If the message transmission occurs before the handoff completes, however, the message will be sent to the previous base station st where it will be forwarded to the new base station $s't'$. Define τ_{st} to be the average time in second for a mobile host to stay at base station st before it leaves. Then $\tau_{st} = H_{st}/M_{st}$, where H_{st} is the total number of mobile hosts at base station st and M_{st} is the average rate of mobile hosts per second moving out of base station st. Given the average handoff time τ_h, the probability that a message destined to base station st is forwarded to base station $s't' \in Adj(st)$ is given by $\sigma_{st} = \tau_h/\tau_{st}|Adj(st)|$.

This should be a valid estimate if an MH migrates independent of when it receives data messages.

Define $q^{out}_{st,s't'}$ to be the forwarding frequency going out of base station st to base station $s't' \in Adj(st)$, q^{out}_{st} to be the forwarding frequency going out of base station st to all neighboring base stations in $Adj(st)$, and q^{out}_{s} to be the forwarding frequency going out of all base stations in virtual cell s to different virtual cells. Then $q^{out}_{st,s't'} = f_f(xy, st)\sigma_{st}$, $q^{out}_{st} = \sum_{s't' \in Adj(st)} q^{out}_{st,s't'}$, and $q^{out}_{s} = \sum_{st} \sum_{s't' \in Adj(st), s' \neq s} q^{out}_{st,s't'}$.

Define $q^{in}_{st,s't'}$ to be the forwarding frequency going out of base station $s't' \in Adj(st)$ into base station st, q^{in}_{st} to be the forwarding frequency going out of all neighboring base stations in the same virtual cell s into base station st, $q^{in}_{\overline{st}}$ to be the forwarding frequency going out of all neighboring base stations in different virtual cells into base station st, and q^{in}_{s} to be the forwarding frequency going out of all neighboring base stations in different virtual cells into all base stations in virtual cell s. Then $q^{in}_{st,s't'} = f_f(xy, s't')\sigma_{s't'}$, $q^{in}_{st} = \sum_{s't' \in Adj(st), s' = s} q^{in}_{st,s't'}$, $q^{in}_{\overline{st}} = \sum_{s't' \in Adj(st), s' \neq s} q^{in}_{st,s't'}$, and $q^{in}_{s} = \sum_{st} q^{in}_{\overline{st}}$.

From the above forwarding frequencies, we can compute the following probabilities used to determine service transition probabilities for the forwarding message classes:

- The probability that a forwarded message at base station st is routed to a different virtual cell is given by $\beta^{inter}_{s} = q^{out}_{s} / \sum_{st} q^{out}_{st}$.
- The probability that a forwarded message at base station st is routed to base station st' in the same virtual cell s is given by $\beta^{intra}_{st'} = q^{in}_{st'} / \sum_{st'} q^{in}_{st'}$.
- The probability that a forwarded message at base station st is routed to the base stations in a different virtual cell s' is given by $\gamma^{inter}_{s'} = q^{in}_{s'} / \sum_{s'} q^{in}_{s'}$.

- The probability that a forwarded message at base station st is routed to base station $s't'$ is given by $\gamma_{s't'} = q^{in}_{s't'} / \sum_{s't'} q^{in}_{s't'}$.

The service transition matrix $P_{x,y,2}$ for the data message class generated at base station xy is given as follows:

- When a data class message is destined to base station xy' in the same virtual cell x, service transition probabilities are as follows:
  - $p_{xy1;xB1} = 1.0$ and $p_{xB1;xy'1} = \alpha_{xy'}/\alpha_x$ for the data class,
  - $p_{xy'1;xB2} = \sigma_{xy'}|Adj(xy')|$, $p_{xB2;xy'2} = (1 - \beta^{inter}_x)\beta^{intra}_{xy'}$, and $p_{xB2;xS2} = \beta^{inter}_x$ for the intra-forwarding class, and
  - $p_{xS2;B13} = 1.0$, $p_{B13;x'S3} = \gamma^{inter}_{x'}$, $p_{x'S3;x'B3} = 1.0$, and $p_{x'B3;x'y'3} = \gamma_{x'y'}$ for the inter-forwarding class.
- When a data class message is destined to base station $x'y'$ in a different virtual cell x', service transition probabilities are as follows:
  - $p_{xy1;xB1} = 1.0$, $p_{xB1;xS1} = 1 - \alpha_x$, $p_{xS1;B11} = 1.0$, $p_{B11;x'S1} = \alpha_{x'}/(1 - \alpha_x)$, $p_{x'S1;x'B1} = 1.0$, and $p_{x'B1;x'y'1} = \alpha_{x'y'}/\alpha_{x'}$ for the data class,
  - $p_{x'y'1;x'B2} = \sigma_{x'y'}|Adj(x'y')|$, $p_{x'B2;x'y'2} = (1 - \beta^{inter}_{x'})\beta^{intra}_{x'y'}$, and $p_{x'B2;x'S2} = \beta^{inter}_{x'}$ for the intra-forwarding class, and
  - $p_{x'S2;B13} = 1.0$, $p_{B13;x'S3} = \gamma^{inter}_{x'}$, $p_{x'S3;x'B3} = 1.0$, and $p_{x'B3;x'y'3} = \gamma_{x'y'}$ for the inter-forwarding class.

Address resolution message classes. Consider the address resolution request, address resolution reply, and address resolution reply forwarding message classes originated from base station xy, i.e., a routing chain $E_{x,y,3}$. Denote h_{BS} as the hit ratio

of the cache table at a base station for a mobile host. The service transition matrix $P_{x,y,3}$ for address resolution message classes originated from base station xy is given as follows:

- $p_{xy1;xB1} = 1 - h_{BS}$ and $p_{xB1;xS1} = 1.0$ for the address resolution request class,
- $p_{xS1;xB2} = 1.0$ and $p_{xB2;xy2} = 1.0$ for the address resolution reply class, and
- $p_{xy2;xB3} = \sigma_{xy}|Adj(xy)|$ and $p_{xB3;xy'3} = 1/|Adj(xy)|$ for the address resolution reply forwarding class.

6.1.4 Traffic Equations

Suppose that e_{ijr} is the average throughput of class r messages through node ij in a routing chain $E_{x,y,z}$. Then for node ij and class r, $(ij, r) \in E_{x,y,z}$, we can write the following traffic equations:

$$\sum_{(kl,s) \in E_{x,y,z}} e_{kls} p_{kls;ijr} + p_{0;ijr} = e_{ijr},$$

where $p_{0;ijr}$ is the probability that an external arrival is for node ij and class r and determined by the rate of external arrivals of class r messages to node ij. Since $p_{0;ijr} > 0$ for some $(ij, r) \in E_{x,y,z}$, $E_{x,y,z}$ is open and so the traffic equations have a unique solution for $\{e_{ijr}\}$.

6.2 Performance Measures

The BCMP theorem states that multiple class queueing networks with the FCFS, PS, IS, and LCFS-PR types of nodes have a product form solution for the steady state joint probability distribution of the node states [26]. Since in an open network

a message sees the network in the steady state when it leaves or enters the network, Little's result can be applied to any given class of messages at a node.

Node ij has a fixed service rate of μ_{ijr} and relative throughput of e_{ijr} for class r messages. The service time of FCFS nodes is independent of the class and so $1/\mu_{ijr} = 1/\mu_{ij}$ for all classes r. The relative throughput e_{ijr} is the average number of times that a class r message visits node ij before leaving the network. Then the total service demand of a class r message at node ij is $D_{ijr} = e_{ijr}/\mu_{ijr}$.

Let η_r be the external arrival rate of each class r. The performance measures we can obtain from the described queueing network model include:

- The throughput of class r messages in the steady state is $T_r = \eta_r$.
- The utilization of node ij by class r messages is $U_{ijr} = \eta_r D_{ijr}$ by Little's result. Thus, the utilization of node ij is given by $U_{ij} = \sum_{r \in R} \eta_r D_{ijr}$, where R is the set of all message classes.
- The mean waiting time of a class r message at node ij for each visit is given by $W_{ijr} = (1/\mu_{ijr})/(1 - U_{ij})$.
- The mean time that a class r message spends at node ij during its stay in the network, i.e., the mean residence time at node ij for class r is given by $Q_{ijr} = e_{ijr} W_{ijr}$.
- The mean time that a class r message spends in the network, i.e., the system response time for class r is given by $Q_r = \sum_{ij \in N} Q_{ijr}$.
- The mean number of class r messages at node ij is $L_{ijr} = \eta_r Q_{ijr}$ by Little's result.

6.3 Performance Evaluations

This section gives the results of computations from the performance measures of the queueing network model. By changing the density, velocity and in-call probability of mobile hosts among mobility and traffic parameters, we evaluate the utilization of various network components and the system response time due to various messages. With the same model parameters we also compare two different virtual cell systems: one deployed according to an initial partition and the other deployed according to an optimal partition with respect to the initial partition. The initial model parameters are listed as follows:

- System parameters

the number of base stations(n)	19
the number of virtual cells(m)	3
base station network service rate(μ_{iB})	45 Mbps
backbone network service rate(μ_{B1})	45 Mbps
base station service rate(μ_{ij})	4.5 Mbps
ARP/Location server service rate(μ_{iS})	4.5 Mbps

- Mobility and traffic parameters

average density of mobile hosts at physical cell $xy(\rho_{xy})$	500 mh/km^2
average velocity of mobile hosts at physical cell $xy(v_{xy})$	15 km/hr
cell radius(R)	1 km
data rate of a wireless channel(b)	2 Kbps
average message length(l)	4 Kbits
Erlangs per mobile host(E)	0.04
average number of messages over a conversation(c)	240
address resolution miss ratio at a mobile host(h_m)	0.5
address resolution hit ratio at a base station(h_{BS})	0.5
probability that a mobile host is powered on	0.5
average handoff time(τ_h)	1 sec

The analysis assumes that the initial density of mobile hosts in each of n physical cells is randomly varied between 325 mh/km^2 and 650 mh/km^2 such that the average

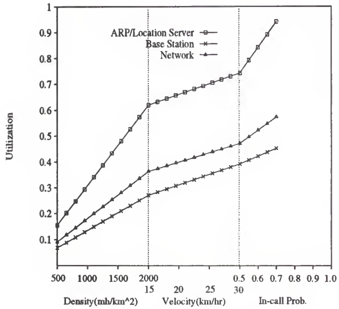


Figure 6.3. The Utilization of the Network Components in the Virtual Cell System When an Initial Partition Is Used

density becomes 500 mh/km^2. The initial velocity of mobile hosts in each of n physical cells is also randomly varied between 5 km/hr and 25 km/hr such that the average velocity becomes 15 km/hr. The initial density of each physical cell is increased with the same ratio at each step such that the average density is raised from 500 mh/km^2 to 2000 mh/km^2. After the average density increased up to 2000 mh/km^2, the average velocity is then increased from 15 km/hr up to 30 km/hr in the same way and then the in-call probability is raised from 0.5 up to 1.0. The base station network service rate μ_{iB} and the backbone network service rate μ_{B1} represent the service rate of the same physical transport network.

Figure 6.3 and Figure 6.4 depict the utilization of the network components in the virtual cell system which is deployed according to an initial partition and an optimal partition, respectively. Since the same physical transport network is used for both base station networks and the backbone network to construct a virtual cell system, the network utilization of Figure 6.3 and Figure 6.4 represents the sum of all

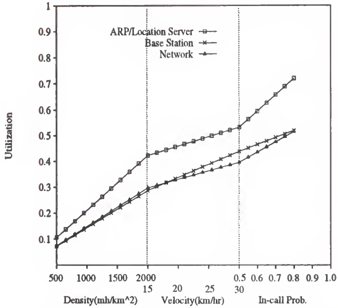


Figure 6.4. The Utilization of the Network Components in the Virtual Cell System When an Optimal Partition Is Used

utilizations for three base station networks and one backbone network. The utilization of the ARP/Location server and base stations represents the average utilization of an ARP/Location server or a base station.

As depicted in Figure 6.3, the high utilization of the ARP/Location server for the initial partition implies that there is a large volume of mobility and data traffic between clusters before the optimization process. It is shown in Figure 6.4 that the inter-cluster traffic is significantly reduced after the optimization process. The less inter-cluster traffic means that the utilizations of the backbone network and base station networks are also reduced. This is due to the fact that for handoff or data transfer operations, intra-cluster traffic involves operations at only one base station network while inter-cluster traffic involves operations at two base station networks and one backbone network.

It is interesting to observe that the average utilization of a base station for the optimal partition is slightly higher than that for the initial partition and it is especially sensitive to the velocity of mobile hosts. This is due to the fact that the optimization process produces as large clusters as possible within the cluster size constraint in order to reduce the total communication cost for the entire system. Consider the impact of handoff multicast operations in terms of cluster size. The number of mobile hosts leaving a base station is determined by the density and velocity of mobile hosts in its physical cell, not the cluster size. For each leaving mobile host, the base station invokes a multicast operation to all other base stations in the same cluster in order to maintain the consistency of the distributed location information of the mobile host. Thus, a handoff multicast operation within a large cluster involves more base stations than that within a small cluster. This increases the total utilization of base stations. When considering only the handoff multicast operation, the best partition would consist of equal-weighted clusters where the weight of a cluster is the product of the number of base stations in the cluster and the number of mobile hosts moving out of physical cells in the cluster. As the velocity of mobile hosts increases, more handoff multicast operations will be needed, which in turn results in more utilization of base stations.

The utilization of the ARP/Location server for the optimal partition is less likely saturated compared to that for the initial partition. However, the average utilization of the ARP/Location server for the optimal partition approaches to approximately 0.72 at the saturation point of the in-call probability 0.8. This means that the ARP/Location server for the largest cluster of the optimal partition is saturated at the in-call probability 0.8. The bottleneck in the ARP/Location server comes from the fact that the destination of data messages are randomly distributed over the entire system and the locality of data traffic patterns is not considered. Thus the

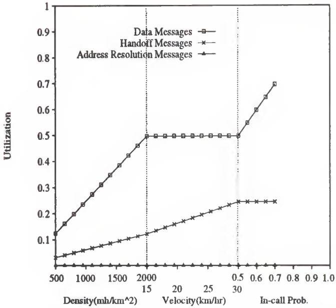


Figure 6.5. The Utilization of the ARP/Location Server for Each Type of Message When an Initial Partition Is Used

ARP/Location server may use a faster processor to resolve it. Adjusting the cluster size constraint in the optimization process could also alleviate the effect of the largest cluster on the utilization of the ARP/Location server. In this analysis, the cluster sizes of the initial partition are $n_1 = 6$, $n_2 = 6$, and $n_3 = 7$, and the cluster sizes of the optimal partition obtained by using the cluster size constraint, $4 < |P_i| < 12$, are $n_1 = 4$, $n_2 = 4$, and $n_3 = 11$.

From Figure 6.5 and Figure 6.6, it should be noted that the significant difference in the utilization of the ARP/Location server between the initial partition and the optimal partition comes from much less inter-cluster data traffic in the optimal partition. Thus, the sensitivity of data messages to the density, velocity, and in-call probability variations dominates the utilization of a network component in the virtual cell system.

For data messages, the increases in the density and in-call probability of mobile hosts directly affect the arrival rate of data messages. However, the increase in the

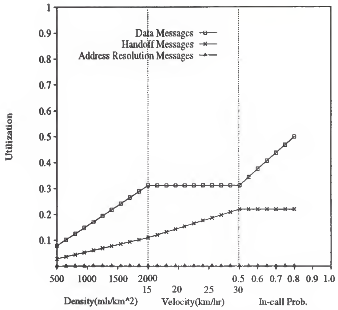


Figure 6.6. The Utilization of the ARP/Location Server for Each Type of Message When an Optimal Partition Is Used

velocity of mobile hosts does not raise the arrival rate but the forwarding rate of data messages. Thus, as the velocity of mobile hosts increases, the utilization of data messages at the ARP/Location server is slightly increased by the inter-cluster forwarding data messages, but it could be negligible compared to that of data transfer class of messages. The utilization of the ARP/Location by handoff messages is more sensitive to the velocity variation than the density variation. The in-call probability does not affect handoff messages because it only raises the arrival rate of data messages. Compared to data and handoff messages, the utilization of the ARP/Location server by address resolution messages could be negligible.

The system response time Q_r for each type of messages are shown in Figure 6.7 and Figure 6.8. In the optimal partition, the system response time for data and address resolution messages is reduced at the cost of slightly higher response time for handoff messages. This is due to the impact of handoff multicast operations on the

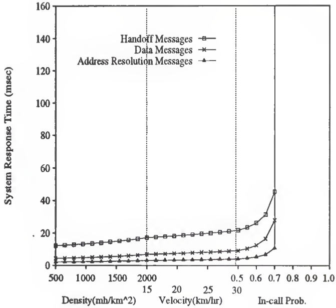


Figure 6.7. The System Response Time for Each Type of Message When an Initial Partition Is Used

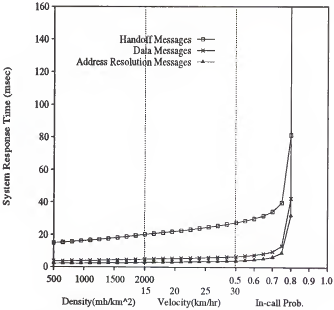


Figure 6.8. The System Response Time for Each Type of Message When an Optimal Partition Is Used

relatively larger cluster size with respect to the cluster size of the initial partition, as described previously. When considering high mobility and data parameters, i.e., 2000 mh/km^2, 30 km/hr, and 0.8 in-call probability of mobile hosts, the mean handoff response time 80 $msec$ is quite acceptable. Given some mobility and data traffic parameters, the analysis shows a trade-off between the data and handoff response times. The much larger volume of data traffic over handoff traffic in mobile data communications reveals that even the small difference in the data response time could improve the overall system performance significantly. To reduce the handoff response time, more strict constraint on the cluster size is needed in the optimization process.

6.4 Chapter Summary

To analyze the performance of the virtual cell system we adopt a BCMP open multiple class queueing network. Since the same type of message may have a different routing behavior depending on which base station belongs to which cluster, a routing chain is defined for each type of message generated by each base station. Both mobility and data traffic patterns among base stations and the topology of the virtual cell system are used to determine service transition probabilities in the queueing network model. With various performance measures such as the utilization of network components in the virtual cell system and the system response time for various types of messages, we have conducted sensitivity analyses of those performance measures as mobility and data traffic parameters vary. We also compared the performance measures of two different virtual cell systems which are one deployed according to an initial partition used for the optimization process and the other deployed an optimal partition with respect to the initial partition, respectively.

CHAPTER 7
CONCLUSIONS

We have presented the network infrastructure of the virtual cell system for the transmission of IP datagrams in mobile computer communications. With the virtual cell concept, physical cells are grouped into larger logical cells where host mobility is shielded from the IP layer. In order to achieve this concept, we have designed the virtual cell protocol which consists of handoff, address resolution, and data transfer modules, based on the distributed hierarchical location information of mobile hosts. It eliminates the need of IP-level protocols for host mobility and provides the flexible coverage area of a virtual cell that can be properly engineered according to mobility and communication patterns among physical cells.

Mobility and communication patterns among physical cells can be represented by the move and find frequencies among base stations because base stations are regarded as traffic sources and destinations from the prospective of the virtual cell system. Given the move and find frequencies among base stations, the optimal deployment of the virtual cell system is equivalent to the problem of finding an optimal partition of disjoint clusters of base stations. For each type of operation, inter-cluster communication is more expensive than intra-cluster communication. The objective is to minimize the total communication cost for a sequence of move and find operations in the entire system. The optimization problem differs from general graph partitioning problems in that it additionally considers the underlying topology of base stations

such as the linear arrangement of base stations in highway cellular systems and the hexagonal mesh arrangement of base stations in cellular systems.

For highway cellular systems, we have showed that the optimization problem is solvable in polynomial time of $O(mn^2)$ by dynamic programming for an arbitrary number of clusters and size of a cluster, where n is the number of base stations and m is the number of clusters in a partition. The algorithm also finds all valid partitions in the same polynomial time if given a constraint on the size of a cluster and the total allowable communication cost for the entire system.

For hexagonal cellular systems, several heuristics are considered using the techniques of interchanging or moving boundary nodes between adjacent clusters. Experimental testing and analysis show that unlike general graph partitioning problems, some heuristics relying on only one of the techniques are no longer useful for the optimization problem. We have developed several heuristics based on the combinations of the techniques of moving or interchanging the boundary nodes between adjacent clusters. These heuristics produce optimal partitions with respect to the initial partition obtained randomly or by centering. The heuristics are compared and shown to behave quite well through experimental testing and analysis.

Finally we have analyzed the performance of the virtual cell system under the assumption that it is deployed according to the topology of the optimal partition such that each cluster corresponds to a virtual cell. The virtual cell system is modeled as a BCMP open multiple class queueing network. In addition to the move and find frequencies among base stations, the topology of the optimal partition is used to determine service transition probabilities and the arrival rates of each type of message. By solving traffic equations of the network model, we obtain the interesting performance measures such as the network response time for each type of message

and the utilization of the base station networks and the backbone network of the virtual cell system.

REFERENCES

[1] John Ioannidis, Dan Duchamp, and Gerald Q. Maguire Jr., "IP-based Protocols for Mobile Internetworking," *ACM SIGCOMM'91*, pp. 235-245, 1991.

[2] Fumio Teraoka, Yasuhiko Yokote, and Mario Tokoro, "A Network Architecture Providing Host Migration Transparency," *ACM SIGCOMM'91*, pp. 209-220, 1991.

[3] Fumio Teraoka, Kim Claffy, and Mario Tokoro, "Design, Implementation, and Evaluation of Virtual Internet Protocol," *IEEE DCS'92*, pp. 170-177, 1992.

[4] Charles Perkins and Yakov Rekhter, "Short-cut Routing for Mobile Hosts," draft RFC, T. J. Watson Research Center, IBM Corp., July 1992.

[5] Charles Perkins and Yakov Rekhter, "Support for Mobility with Connectionless Network Layer Protocols," draft RFC, T. J. Watson Research Center, IBM Corp., Jan. 1993.

[6] Hiromi Wada, Tatsuya Ohnishi, and Brian Marsh, "Packet Forwarding for Mobile Hosts," draft RFC, Matsushita Corp., Nov. 1992.

[7] Masatoshi Kawarasaki and Bijan Jabbari, "B-ISDN Architecture and Protocol," *IEEE J. Select. Areas Commun.*, Vol. 9, No. 9, pp. 1405-1415, December 1991.

[8] Jean-Yves Le Boudec, "The Asynchronous Transfer Mode: a Tutorial," *Computer Networks and ISDN Systems*, Vol. 24, pp. 279-309, 1992.

[9] David M. Piscitello and Michael Kramer, "Internetworking Using Switched Multi-megabit Data Service in TCP/IP Environments," *Computer Networks and ISDN Systems*, pp. 62-71, 1991.

[10] F. R. Dix, M. Kelly, and R. W. Klessig, "Access to a Public Switched Multi-megabit Data Service Offering," *Computer Networks and ISDN Systems*, pp. 46-61, 1991.

[11] David J. Goodman, Gregory P. Pollini, and Kathleen S. Meier-Hellstern, "Network Control for Wireless Communications," *IEEE Commun. Mag.*, pp. 116-124, 1992.

[12] Kathleen S. Meier-Hellstern, Gregory P. Pollini, and David J. Goodman, "A Wireless Service for the IEEE 802.6 Metropolitan Area Network," *IEEE GLOBECOM'91*, pp. 1964-1968, 1991.

[13] Michael R. Garey and David S. Johnson, *Computers and Interactability – A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, 1979.

[14] Oliver Goldschmidt and Dorit S. Hochbaum, "Polynomial Algorithm for the k-cut Problem," *29th Annual Symp. on Foundation of Computer Science*, pp. 444-451, 1988.

[15] Brenda S. Baker, "Approximation Algorithms for NP-Complete Problems on Planar Graphs," *Proc. Symposium on Foundations of Computer Science*, pp. 265-273, 1983.

[16] Thang Nguyen Bui and Andrew Peck, "Partitioning Planar Graphs," *SIAM J. Comput.*, Vol. 21, No. 2, pp. 203-215, April 1992.

[17] Susan B. Davidson, "Consistency in Partitioned Networks," *ACM Computing Surveys*, Vol. 17, No. 3, pp. 341-370, September 1985.

[18] Jo-Mei Chang and N. F. Maxemchuk, "Reliable Broadcast Protocol," *ACM Trans. on Computer Systems*, Vol. 2, No. 3, pp. 251-273, August 1984.

[19] P. M. Melliar-Smith, Louise E. Moser, and Vivek Agrawala, "Broadcast Protocols for Distributed Systems," Vol. 1, No. 1, pp. 17-25, 1990.

[20] Stanley Chia, "The Universal Mobile Telecommunication System," *IEEE Communications Magazine*, pp. 54-62, December 1992.

[21] Ramon Caceres, Peter B. Danzig, Sugih Jamin, and Danny J. Mitzel, "Characteristics of Wide-Area TCP/IP Conversations," *ACM SIGCOMM'91*, pp. 101-112, 1991.

[22] K. M. Khalil, J. C. Hand, and M. Mariswamy, "Analysis and Traffic Characterization of a Wide Area Network," *IEEE ICC'93*, pp. 1829-1835, 1993.

[23] Kyungshik Lim and Yann-Hang Lee, "Optimal Partitioning of Heterogeneous Traffic Sources in Highway Cellular Systems," *Personal, Indoor and Mobile Radio Communications PIMRC'94*, The Hague, The Netherlands, Sep. 1994.

[24] Kyungshik Lim and Yann-Hang Lee, "Heuristics for Multiway Partitioning in Hexagonal Cellular Systems," Submitted to ICC'95.

[25] R. Thomas, H. Gilbert, and G. Mazziotto, "Influence of Moving of the Mobile Stations on the Performance of a Radio Mobile Cellular Network," *Proc. 3rd Nordic Seminar*, Paper 9.4, Copenhagen, Denmark, Sep. 1988.

[26] Forest Baskett, K. Mani Chandy, Richard R. Muntz, and Fernando G. Palacios, "Open, Closed, and Mixed Networks of Queues with Different Classes of Customers," *J. ACM*, Vol. 22, No. 2, April 1975, pp. 248-260.

[27] Max M.-K. Liu, "Base Station Networking in Personal Communications," *IEEE GLOBECOM'91*, pp. 1912-1916, 1991.

[28] David J. Goodman, "Packet Transmission and Switching in Advanced Wireless Information Networks," *IEEE ICC'90*, pp. 1473-1478, 1990.

[29] D. Piscitello and J. Lawrence, "The Transmission of IP Datagrams over the SMDS Service," *RFC* 1209, Bell Communications Research, Mar. 1991.

[30] George H. Clapp, "LAN Interconnection Across SMDS," *IEEE Network Magazine*, pp. 25-32, September 1991.

[31] Davide Grillo, R. J. Gerard MacNamee, and Behrooz Rashidzadeh, "Towards Third Generation Mobile Systems: a European Possible Transition Path," *Computer Networks and ISDN Systems*, Vol. 25, pp. 947-961, 1993.

[32] Samuel Sheng, Anantha Chandrakasan, and R. W. Brodersen, "A Portable Multimedia Terminal," *IEEE Communications Magazine*, pp. 64-75, December 1992.

[33] Efstathios D. Sykas and Michael E. Theologou, "Numbering and Addressing in IBCN for Mobile Communications," *Proc. of the IEEE*, Vol. 79, No. 2, pp. 230-241, February 1991.

[34] Moe Rahnema, "Overview of the GSM System and Protocol Architecture," *IEEE Communications Magazine*, pp. 92-100, April 1993.

[35] Hans de Boer, Marcel Meijer, and Evert Buitenwerf, "Network Aspects for the Third Generation Mobiles," *IEEE GLOBECOM'91*, pp. 1517-1522, 1991.

[36] Baruch Awerbuch and David Peleg, "Concurrent Online Tracking of Mobile Users," *ACM SIGCOM'91*, pp. 221-233, 1991.

[37] Amotz Bar-Noy and Ilan Kessler, "Tracking Mobile Users in Wireless Communications Networks," *IEEE INFOCOM'93*, pp. 1232-1239, 1993.

[38] T. Imielinski and B. R. Badrinath, "Querying Locations in Wireless Environments," *Third Workshop on Third Generation Wireless Information Networks*, April 1992.

[39] Andrew Schmidt and Roy Campbell, "Internet Protocol Traffic Analysis with Applications for ATM Switch Design," *ACM SIGCOMM'92*, pp. 39-52, 1992.

[40] Baruch Awerbuch and David Peleg, "Sparse Partitions," *IEEE 31th Annual Symp. on Foundations of Computer Science*, pp. 503-513, 1990.

[41] E. Dahlhaus, D. S. Johnson, and C. H. Papadimitriou, "The Complexity of Multiway Cuts," *Proc. of the 24th Annual ACM Symp. on the Theory of Computing*, pp. 241-251, 1992.

[42] Edward Minieka, *Optimization Algorithms for Networks and Graphs*, Marcel Dekker, Inc., 1978.

[43] L. Caccetta, "Graph Theory in Network Design and Analysis," *Recent Studies in Graph Theory*, ed. V. R. Kulli, Vishwa International Publications, 1989.

[44] Kyungshik Lim and Yann-Hang Lee, "Virtual Cell in Mobile Computer Communications," *International Conference on Computer Communications Networks ICCCN'94*, Sep. 1994.

[45] Kyungshik Lim and Yann-Hang Lee, "A Performance Analysis of the Virtual Cell System in Mobile Computer Communications," Submitted to APCC'95.

[46] Peter G. Harrison and Naresh M. Patel, *Performance Modeling of Communication Networks and Computer Architectures*, Addison-Wesley, 1993.

BIOGRAPHICAL SKETCH

Kyungshik Lim was born in Taegu, Korea, in 1959. He received a B.S. degree in electronics engineering from Kyungpook National University, Taegu, Korea, in 1982 and an M.S. degree in computer science from the Korea Advanced Institute of Science and Technology, Seoul, Korea, in 1985. He is a senior member of the research staff of the Electronics and Telecommunications Research Institute, Taejon, Korea, which he joined in 1985. Since the fall semester of 1990, he has been in the Ph.D. program of the Computer and Information Sciences Department at the University of Florida, serving as a teaching or a research assistant. His research interests include mobile computer communications, wireless networks, high-speed communications networks, and parallel and distributed systems.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Yann-Hang Lee, Chairman
Associate Professor of Computer
and Information Sciences

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Randy Chow
Professor of Computer
and Information Sciences

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Richard E. Newman-Wolfe
Assistant Professor of Computer
and Information Sciences

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Panos Livadas
Assistant Professor of Computer
and Information Sciences

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Scott L. Miller

Scott L. Miller
Associate Professor of Electrical Engineering

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

December 1994

Winfred M. Phillips
Dean, College of Engineering

Karen A. Holbrook
Dean, Graduate School